ISSN 2713-3192
DOI 10.15622/ia.2026.25.2
http://ia.spcras.ru

ТОМ 25 № 2

ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ

INFORMATICS AND AUTOMATION

СПб ФИЦ РАН

Санкт-Петербург
2026

INFORMATICS AND AUTOMATION

Volume 25 № 2, 2026

A scientific and educational journal primarily specializing in computer science, automation, robotics, applied mathematics and interdisciplinary research
Founded in 2002

Frequency
6 times a year

Founder and Publisher:
St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS)

Editorial Office and Publisher Address:
39, 14th Line V.O., St.Petersburg, 199178, Russia
e-mail: ia@spcras.ru, website: http://ia.spcras.ru

Editor-in-Chief:
A. L. Ronzhin, Dr. Sci., Prof., St. Petersburg, Russia

Managing Editor: V. A. Ganicheva
Translator: Ya. N. Berezina
Art Editor: Yu. S. Cherepanova

The journal is registered with the Federal Service for Supervision of Communications, Information Technology, and Mass Media,
Registration Certificate (registration number) ПИ № ФС77-79228 dated September 25, 2020

The journal is indexed in the Scopus international database

The journal is included in the «List of Leading Peer-Reviewed Scientific Journals and Publications, in which the main scientific results of a dissertation for the degree of Doctor and Candidate of Sciences must be published»

The journal is published under the scientific and methodological guidance of the Department of Nanotechnology and Information Technologies of the Russian Academy of Sciences

ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ

Том 25 № 2, 2026

Научный, научно-образовательный журнал с базовой специализацией
в области информатики, автоматизации, робототехники, прикладной математики
и междисциплинарных исследований.
Журнал основан в 2002 году

Периодичность
6 раз в год

Учредитель и издатель:
Федеральное государственное бюджетное учреждение науки
«Санкт-Петербургский Федеральный исследовательский центр Российской академии наук»
(СПб ФИЦ РАН)

Адрес редакции и издателя:
Россия, 199178, г. Санкт-Петербург, 4-я линия В.О., д. 39, лит. А
e-mail: ia@spcras.ru, сайт: http://ia.spcras.ru

Журнал зарегистрирован Федеральной службой по надзору в сфере связи,
информационных технологий и массовых коммуникаций,
свидетельство о регистрации (регистрационный номер) ПИ № ФС77-79228
от 25 сентября 2020 г.

Журнал индексируется в международной базе данных Scopus

Журнал входит в «Перечень ведущих рецензируемых научных журналов и изданий,
в которых должны быть опубликованы основные научные результаты диссертации
на соискание ученой степени доктора и кандидата наук»

Журнал выпускается при научно-методическом руководстве Отделения нанотехнологий
и информационных технологий Российской академии наук

CONTENTS

Artificial Intelligence, Knowledge and Data Engineering

A. Vyatkin, A. Poptsov, V. Oliseenko, M. Abramov
RESEARCH ON THE APPLICABILITY OF MATRIX FACTORIZATION FOR RANKING LARGE LANGUAGE MODELS 275

A. Tolstykh, A. Golubinskiy
DATA MINING BASED ON DEEP REINFORCEMENT LEARNING FOR PREDICTION OF OPERATING FREQUENCIES AND BANDS IN A COGNITIVE RADIO SYSTEM 301

A. Mironov, A. Saenko, E. Fomina
DEVELOPMENT OF A DIGITAL TWIN ARCHITECTURE FOR AN AQUATIC ECOSYSTEM TO PRESERVE ENVIRONMENTAL SUSTAINABILITY 323

A. Vorobev, G. Vorobeva
A CONTEXT-DEPENDENT METHOD FOR ADAPTIVE TUNING OF PARAMETERS OF AUTOREGRESSIVE MODELS FOR NON-STATIONARY TIME SERIES 352

Nguyen Viet Hung, Phi Dinh Huynh, Ma Van Tung, Nguyen Van Vu, Nguyen Phu Dat
CLVM: A HYBRID DEEP LEARNING FRAMEWORK FOR CONTACTLESS VIRTUAL MOUSE CONTROL 378

M.A. Atmane, H. Zatla, B. Tolbi, S.F. Nouar, M. Bouhamama
INTELLIGENT FAULT DETECTION AND ISOLATION BASED ON NARX NEURAL NETWORKS 410

Hoan Viet Bui, Nghia Duc Tran, To-Hieu Dao, Hoang Quang Trung, Pham Vu Kien, Nguyen Van Thang, Duc-Tan Tran.
IDRRS: IOT INERTIAL DEVICE FOR REAL-TIME ROAD SURFACE CLASSIFICATION AND POSITION ESTIMATION ENHANCEMENT 445

Mathematical Modeling and Applied Mathematics

R. Rogulin
TWO-ECHELON TRANSPORT SYSTEM MODEL AND ANT COLONY OPTIMIZATION ALGORITHM: SCALABILITY ANALYSIS OF COMPUTATIONAL SOLUTIONS 478

V. Zinov
HOLES PROCESSING IN VORONOI DIAGRAM WITH RECTANGULAR SITES FOR THE COMPARATIVE ANALYSIS OF SLAB DEFORMATIONS 517

E. Trushkin, V. Freyman
A PREDICTIVE METHOD OF RESOURCE ALLOCATION IN COMPUTING SYSTEMS BASED ON A MULTICRITERIA DECISION-MAKING MODEL 568

СОДЕРЖАНИЕ

Искусственный интеллект, инженерия данных и знаний

А.А. Вяткин, А.В. Попцов, В.Д. Олисеенко, М.В. Абрамов
ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ МЕТОДА МАТРИЧНОЙ ФАКТОРИЗАЦИИ ДЛЯ РАНЖИРОВАНИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ 275

А.А. Толстых, А.Н. Голубинский
ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ НА БАЗЕ ГЛУБОКОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ ПРОГНОЗА РАБОЧИХ ЧАСТОТ И ПОЛОС В СИСТЕМЕ КОГНИТИВНОГО РАДИО 301

А.С. Миронов, А.А. Саенко, Е.С. Фомина
РАЗРАБОТКА АРХИТЕКТУРЫ ЦИФРОВОГО ДВОЙНИКА ЭКОСИСТЕМЫ ВОДНОГО ОБЪЕКТА ДЛЯ СОХРАНЕНИЯ ЭКОЛОГИЧЕСКОЙ УСТОЙЧИВОСТИ 323

А.В. Воробьев, Г.Р. Воробьева
КОНТЕКСТНО-ЗАВИСИМЫЙ МЕТОД АДАПТИВНОЙ НАСТРОЙКИ ПАРАМЕТРОВ АВТОРЕГРЕССИОННЫХ МОДЕЛЕЙ ДЛЯ НЕСТАЦИОНАРНЫХ ВРЕМЕННЫХ РЯДОВ 352

Н.В. Хунг, Ф.Д. Хуинь, М.В. Тунг, Н.В. Ву, Н.Ф. Дат
CLVM: ГИБРИДНАЯ МОДЕЛЬ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ БЕСКОНТАКТНОГО УПРАВЛЕНИЯ ВИРТУАЛЬНОЙ МЫШЬЮ 378

М.А. Атмане, Х. Затла, Б. Толби, С. Ф. Нуар, М. Бухамама
ИНТЕЛЛЕКТУАЛЬНОЕ ОБНАРУЖЕНИЕ И ИЗОЛЯЦИЯ НЕИСПРАВНОСТЕЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ NARX 410

Вьет-Хоан Буй, Дук-Нгиа Тран, Ту-Хиеу Дао, Хоанг Куанг Чунг, Фам Ву Киен, Нгуен Ван Тханг, Дук-Тан Тран.
IDRRS: ИНЕРЦИАЛЬНОЕ IOT-УСТРОЙСТВО ДЛЯ КЛАССИФИКАЦИИ ДОРОЖНОГО ПОКРЫТИЯ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ И ПОВЫШЕНИЯ ТОЧНОСТИ ОЦЕНКИ МЕСТОПОЛОЖЕНИЯ 445

Математическое моделирование и прикладная математика

Р.С. Рогулин
ДВУХЭШЕЛОННАЯ МОДЕЛЬ ТРАНСПОРТНОЙ СИСТЕМЫ И МУРАВЬИНЫЙ АЛГОРИТМ: АНАЛИЗ МАСШТАБИРУЕМОСТИ ВЫЧИСЛИТЕЛЬНЫХ РЕШЕНИЙ 478

В.И. Зинов
ОБРАБОТКА ОТВЕРСТИЙ В ДИАГРАММЕ ВОРОНОГО С ПРЯМОУГОЛЬНЫМИ ОСНОВАНИЯМИ ДЛЯ СРАВНИТЕЛЬНОГО АНАЛИЗА ДЕФОРМАЦИЙ В ПЛИТАХ 517

Е.С. Трушкин, В.И. Фрейман
ПРЕДСКАЗАТЕЛЬНЫЙ МЕТОД РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ НА ОСНОВЕ МНОГОКРИТЕРИАЛЬНОЙ МОДЕЛИ ПРИНЯТИЯ РЕШЕНИЙ 568

УДК 004.056 **DOI** 10.15622/ia.25.2.1

А.А. Вяткин, А.В. Попцов, В.Д. Олисеенко, М.В. Абрамов

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ МЕТОДА МАТРИЧНОЙ ФАКТОРИЗАЦИИ ДЛЯ РАНЖИРОВАНИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Вяткин А.А., Попцов А.В., Олисеенко В.Д., Абрамов М.В. **Исследование применимости метода матричной факторизации для ранжирования больших языковых моделей.**

Аннотация. В последние годы широкое применение в области финансов получили большие языковые модели (англ. Large Language Models, LLM). Прямое сравнение таких моделей может быть затруднено, так как наборы данных и сами LLM могут быть закрыты, а параметры при оценке могут отличаться. В работе для задачи заполнения неизвестных метрик предлагается использование метода матричной факторизации из рекомендательных систем, изначально созданного для прогнозирования предпочтений пользователей. Целью работы является оценка применимости матричной факторизации для предсказания метрик качества LLM на финансовых задачах, а также разработка метода ранжирования LLM на основе агрегации метрик качества. Проводится эксперимент по применению матричной факторизации на собранных из научных исследований данных о 34 LLM и 42 финансовых наборах данных. Усредненная MAE метода на всех запусках составляет 0.07 на тестовом наборе данных. Верхние позиции в рейтинге занимают модели DeepSeek R1, OpenAI GPT-4o, OpenAI o1-mini, Fin-R1, Claude 3.5 Sonnet. Двумя способами исследуется влияние ошибки прогнозирования на итоговые предсказания: при помощи MAE и метода Монте Карло. Анализируются полученные результаты, основными выводами которых являются: а) метод матричной факторизации может быть применен для прогнозирования неизвестных значений метрик моделей на наборах данных; б) ведущие большие языковые модели сблизились в оценке настолько, что невозможно выявить явного лидера; в) большие ошибки предсказания позволяют выявить специфические особенности моделей на конкретных задачах. Представленный метод ранжирования способен упростить выбор подходящей модели для финансовых задач.

Ключевые слова: большие языковые модели, оценка качества моделей, матричная факторизация, финансовая сфера.

1. Введение. По мере развития больших языковых моделей (*англ. Large Language Models*, LLM) расширяется охват научных и промышленных областей, в которых они все более успешно внедряются [1]. Одной из сфер, где большие языковые модели могут дать существенный социальный и экономический эффект, является финансовая сфера [2]. В данной сфере LLM могут выступать автономными чат-ботами первой линии технической поддержки или в качестве помощников для второй и остальных линий поддержки, значительно сокращая время отклика и рабочую нагрузку, положительно влияя на качество обслуживания клиентов и клиентоцентричность [3]. Кроме того, LLM могут использоваться в задачах, связанных с обработкой большого количества финансовой документации [4], выявлением мошеннических операций [5],

кредитным скорингом [6], прогнозированием котировок на финансовых рынках [7] и др.

За последние несколько лет разработано большое количество различных LLM [8], и предпринимаются попытки сравнить их как с точки зрения архитектуры, так и качества на разных задачах [9]. Для такого сравнения используются наборы данных и бенчмарки, которые проверяют способности LLM выполнять различные действия. Примерами таких задач в банковской сфере могут выступать: ответы на вопросы по финансовым документам [10], извлечение информации из финансовых текстов [11], анализ тональности финансовых новостей [12], суммаризация финансовой документации [13] и др.

При сравнении моделей классическим решением является использование единого набора данных, как предложено в [14]. В некоторых ситуациях такое сравнение затруднено, поскольку наборы данных и сами LLM могут быть закрыты, а методики оценки, параметры и условия тестирования могут значительно различаться [15]. Поэтому актуальным остается использование других способов сравнения моделей, таких как методы прогнозирования неизвестных значений метрик, например, из области рекомендательных систем. Одним из таких методов является матричная факторизация [16], постановка задачи которой может легко быть применена к LLM и наборам данных для их тестирования. Получающиеся же в результате табличные данные о производительности моделей, в том числе предсказанной, позволяют ранжировать модели и в этом смысле сравнить их.

Целью данной работы является оценка применимости матричной факторизации для предсказания метрик качества LLM на финансовых задачах, а также разработка метода ранжирования LLM на основе агрегации метрик качества. Теоретическая значимость работы заключается в применении известного метода матричной факторизации в новой задаче – оценке LLM в финансовой области для ранжирования моделей в условиях нехватки данных. Практическая значимость состоит в том, что используемый метод позволяет ускорить выбор подходящей для финансовых задач LLM, при отсутствии возможности проведения прямого сравнения на существующих наборах данных.

2. Обзор литературы. В первой части данного раздела рассмотрены работы, использующие методы машинного обучения, изначально разработанные для рекомендательных систем, но применяемые для задач прогнозирования производительности LLM.

Во второй части описаны наборы данных, используемые далее для проведения эксперимента.

Использование методов машинного обучения. Идея применения методов из области рекомендательных систем, таких как матричная факторизация или коллаборативная фильтрация, для прогнозирования оценки качества LLM находит отражение в современных исследованиях [17 – 19]. Это связано с необходимостью оценки качества моделей в условиях, когда прямое их тестирование на всех задачах становится вычислительно затратным, непрактичным или вовсе невозможным в силу закрытости некоторых моделей и бенчмарков [20]. При использовании указанных методов данные представляются в виде матрицы, где строки соответствуют LLM, а столбцы – задачам или наборам данных, а матричная факторизация или коллаборативная фильтрация используется для прогнозирования пропущенных значений в матрице.

В исследовании [21] изучается, насколько предсказуема оценка качества LLM в зависимости от масштаба обучения, который включает в себя увеличение вычислительных ресурсов, количества параметров модели и объема данных для тренировки. Авторы показывают, что для отдельных задач прогнозирование затруднено. Но следует обратить внимание, что общее качество, на больших и разнородных бенчмарках достаточно хорошо предсказывается с помощью гладкой функции потерь, основанной на законах масштабирования.

В работе [17] исследуется метод, позволяющий избежать необходимости проводить дорогостоящие тесты моделей на каждом наборе данных. Предлагается рассматривать задачу предсказания неизвестных значений метрик матрицы как задачу прогнозирования. В матрице строки соответствуют различным большим визуально-языковым моделям, а столбцы – наборам данных. Используя вероятностную матричную факторизацию с цепями Маркова Монте-Карло, авторы демонстрируют значительно лучшие результаты по сравнению с базовыми подходами использования средних значений метрик по моделям и наборам данных (1.5–2 кратное увеличение), в особенности, когда доля тестовой выборки меньше 90%. Таким образом, использование подходов, основанных на применении матричной факторизации, представляется целесообразным, по крайней мере относительно упомянутых выше базовых подходов.

Другой подход, решающий проблему прогнозирования неизвестных метрик, основанный на идеях коллаборативной фильтрации, представлен в исследовании [18]. Здесь для предсказания пропущенных значений также используется матрица с данными

о качестве различных моделей, но с добавлением внешних факторов, описывающих модели и задачи. Этот метод, названный «Collaborative Performance Prediction», позволяет анализировать важность ранее не учитываемых различных факторов и демонстрирует на собранной авторами матрице точность 45%.

В статье [19] для заполнения неизвестных метрик предлагается двухэтапный метод, в котором сначала выбираются наиболее репрезентативные примеры для оценки, а затем, на основе результатов этой выборки, прогнозируется общее качество. Авторы демонстрируют, что их подход превосходит 4 базовых метода, демонстрируя наименьшую среднюю абсолютную ошибку (MAE).

Существующие исследования показывают, что научное сообщество активно разрабатывает методы преодоления проблемы невозможности сравнения моделей при оценке LLM. Тем не менее, представленные выше работы опираются на наборы данных из самых разных областей. В отличие от них, данное исследование полностью сосредоточено на финансовом домене.

Использованные в эксперименте работы. Для применения матричной факторизации было необходимо собрать таблицу со значениями метрик моделей на наборах данных из разных работ. Агрегирование результатов оценки качества LLM из различных источников было выполнено посредством анализа научной литературы. Цель состояла в том, чтобы сформировать единую матрицу данных, где строки соответствуют исследуемым языковым моделям, а столбцы – наборам данных. Каждая ячейка такой матрицы содержит численное значение качества конкретной модели на конкретном задании (наборе данных), измеренное определенной метрикой. Полученная матрица по своей природе является разреженной, поскольку ни одна из публикаций не содержит исчерпывающей оценки всех моделей по всем возможным задачам.

Основными критериями отбора используемых публикаций являлись:

- наличие таблицы со значениями метрик современных моделей из различных финансовых наборов данных;
- наличие пересекающихся моделей или наборов данных с другими работами;
- совпадение значений метрик с другими рассматриваемыми работами (так, в силу невозможности сопоставления, не включались работы, в которых значения метрик моделей на наборах данных отличались от других работ).

После отбора публикаций, отвечающих озвученным критериям, основой для построения используемой далее сводной матрицы послужили следующие четыре работы для анализа.

В работе [22] представлен FLaME – комплексный набор для оценки языковых моделей на финансовых задачах, включающий 20 основных наборов данных для задач обработки естественного языка в финансовой сфере: FinQA, ConvFinQA, TAT-QA, ECTSum, EDTSum [14], FiNER-ORD, FinRED, REFinD, FNXL, FinEntity, SubjECTive-QA, FiQA, FPB, NumClaim, Banking77, FinBench, Headline, FOMC, FinCausal-SC. В рамках исследования используется таксономия на основе сценариев для классификации данных и приводятся значения метрик для 23 моделей (включая GPT-4, Llama-2, BloombergGPT, Gemini, Claude).

В работе [23] представлен набор открытых моделей Open-FinLLMs, проходящих оценку по 14 финансовым задачам из 30 наборов данных. Семейство включает модель FinLLaMA на архитектуре LLaMA3 (дообучена на 52 млрд. токенов), диалоговую модель FinLLaMA-Instruct (настроена на 573 тыс. инструкций) и мультимодальную FinLLaVA (1,43 млн. пар «изображение-текст»), для которых приведены значения метрик (FinLLaMA, FinLLaMA-Instruct, FinLLaVA, LLaMA3, LLaMA3.1).

В работе [24] представлена модель Fin-R1, специально созданная для финансовых рассуждений на основе Qwen2.5-7B-Instruct с использованием обучения с подкреплением. Вместе с моделью был разработан набор данных Fin-R1-Data, содержащий около 60 091 полных цепочек мыслей (Chain-of-Thought) для различных сценариев. Представлены метрики на 5 наборах данных для 10 моделей (основные: Fin-R1, Qwen2.5-7B-Instruct, DeepSeek-R1).

В работе [25] представлена модель KAI-GPT, позиционируемая как первая LLM для банковской отрасли, созданная на базе Pythia-Chat-Base-7B. Для оценки были созданы два новых набора данных: KAI-GPT Evaluation Set #1 (вопросы экспертов с ответами с сайта банка) и KAI-GPT Evaluation Set #2 (краудсорсинг вопросов с форума). Представлены метрики для двух моделей (KAI-GPT, Pythia-Chat-Base-7B). Данное исследование включено в рассмотрение, так как закрытость моделей и данных позволяет посмотреть на работу матричной факторизации на таком типе данных.

Некоторые работы (как, например, Fino1 [26]) не удалось включить в данное исследование, так как значения метрик моделей на наборах данных не совпадают со значениями ни одной другой статьи, что делает невозможным сопоставление. Дополнительно, добавление

новых данных в матрицу увеличило бы ее разреженность еще сильнее, превысив порог, при котором резко падает качество, как показано далее.

3. Постановка задачи. Метод матричной факторизации используется при построении рекомендательных систем [16] и позволяет прогнозировать неизвестные рейтинги товаров/предметов для пользователей. Если в процессе построения матричной факторизации рассматривать наборы данных в качестве «пользователей», а LLM – в качестве «товаров», то данный метод позволит определить неизвестные оценки пользователей-наборов данных по отношению к товарам-моделям. В данной постановке задачи, пользователи-наборы данных будут отдавать предпочтение лучшим товарам-моделям и поэтому ставить им большие оценки-метрики. Метод позволяет учесть скрытые факторы, соответствующие моделям и наборам данных, и с их помощью определять неизвестные рейтинги. Его основа – разложение матрицы метрик $R^{N_u \times N_i}$ на произведение двух матриц $H^{N_u \times N_f}$ и $W^{N_f \times N_i}$:

$$R = HW,$$

где N_u – количество наборов данных, N_i – количество LLM, и N_f – количество скрытых факторов. Прогнозируемая в таком случае оценка $r'_{u,i}$ для набора данных u и модели i рассчитывается следующим образом [16]:

$$r'_{u,i} = \sum_{f=0}^{N_f} H_{u,f} W_{f,i}.$$

Подобная оценка позволяет учесть явные числовые метрики, однако для дополнительного учета неявных взаимодействий используется усовершенствованная формула расчета прогнозируемых значений, которая и будет впоследствии использована для предсказания значений метрик моделей [16]:

$$r'_{u,i} = \mu + b_i + b_u + \sum_{f=0}^{N_f} H_{u,f} W_{f,i}, \quad (1)$$

где μ, b_i и b_u – соответственно общая средняя метрика, а также наблюдаемые отклонения, относящиеся к модели i или набору данных u. В результате, чтобы определить параметры H, W, все b_i и b_u (обозначенные как b_*), система минимизирует регуляризованную квадратичную ошибку:

$$min_{H,W,b_*}\left[\sum_{(u,i)\in K}\left(r_{u,i}-r'_{u,i}\right)^2+\lambda\left(b_u^2+b_i^2+\left|\left|H_{u,*}\right|\right|^2+\left|\left|W_{*,i}\right|\right|^2\right)\right], \quad (2)$$

где $r_{u,i}$ – действительная метрика для набора данных u и модели i, K – набор (u,i), для которых известна $r_{u,i}$ (обучающий набор), $H_{u,*}$ и $W_{*,i}$ – векторы, соответствующие набору данных u и LLM i соответственно, $||\cdot||$ – норма (например, Евклидова), λ – параметр регуляризации [16].

4. Эксперимент. В данном разделе представлено детальное описание эксперимента по предсказанию неизвестных значений метрик больших языковых моделей в финансовой области с использованием метода матричной факторизации. Первым шагом является сбор данных о метриках из различных научных работ, за которым следует их подготовка: фильтрация и формирование итоговой матрицы «модель-набор данных». Далее проводится предварительный анализ влияния разреженности данных на точность прогнозов. Основная часть посвящена непосредственному применению матричной факторизации, включая подбор гиперпараметров для минимизации ошибки прогнозирования. Вводится специальная агрегированная метрика для обеспечения возможности комплексного ранжирования моделей. Для оценки надежности полученных результатов и итогового рейтинга проводится анализ неопределенности прогнозов с использованием двух различных подходов, включая разброс по MAE и метод Монте-Карло.

Подготовка данных. На основе перечисленных в разделе **2** исследований были собраны метрики по 42 наборам данных. Их можно разбить на 9 категорий в зависимости от типа задачи. Предоставим краткое описание общих категорий использованных наборов данных.

Извлечение информации. Категория, включающая наборы данных, предназначенных для извлечения структурированной информации из неструктурированного текста. Наборы данных: FiNER-ORD [11], FinRED [27], REFinD [28], FNXL (Financial Numeric Extreme Labeling) [29], Headlines [30], Mergers and Acquisition (M&A) [30].

Ответы на вопросы. Категория наборов данных, проверяющих способности моделей предоставлять ответы на вопросы, иногда по заданному контексту. Наборы данных: SubjECTive-QA [32], FinQA [10], TAT-QA (Tabular and Textual Question Answering) [33], ConvFinQA (Conversational Finance Question Answering) [34], RegulationsQA [23], Evaluation Set №1 [25], Evaluation Set №2 [25].

Анализ тональности. Категория, содержащая наборы данных, оценивающих способности моделей определять эмоциональную окраску

текста. Наборы данных: FinEntity [12], Financial Phrase Bank (FPB) [35], Targeted Sentiment Analysis (TSA) [36], The Twitter Financial News (TFNS) [37].

Классификация. Группа наборов данных, предназначенных для классификации текстов по заранее заданным категориям. Наборы данных: Banking77 [38], FinCausal-SC [39], Numerical Claim Detection Dataset (NC) [40], FinArg-AUC [41].

Обнаружение мошенничества. Категория, объединяющая наборы данных, направленных на выявление мошеннических операций в финансовых транзакциях и страховых исках. Наборы данных: PortoSeguro [6], travelinsurance [6], Credit Card Fraud (ccf) [6], ccFraud [6].

Суммаризация. Категория наборов данных, предназначенных для создания кратких изложений длинных финансовых текстов и документов. Наборы данных: ECTSum [13], EDTSum [14].

Работа с числами. Категория наборов данных, проверяющих способности моделей производить числовые рассуждения и решать математические задачи в финансовом контексте. Наборы данных: KnowledgeMath [42], DocMath-Eval [43], MC [23].

Прогнозирование, скоринг и оценка. Категория, содержащая наборы данных для проверки возможностей LLM предсказывать различные финансовые события, такие как дефолт, банкротство, финансовые затруднения. Наборы данных: FinBench [44], German (German Credit Data) [45], Australian (Australian Credit Approval) [46], LendingClub [6], polish (Polish Companies Bankruptcy) [6], taiwan (Taiwan Economic Journal dataset) [6].

Общая оценка моделей. Категория, объединяющая наборы данных для оценки общих знаний моделей и способностей в финансовой сфере. Наборы данных: Finance-Instruct-500K [47], Abbreviation [23], Ant_Finance [48], Federal Open Market Committee [49].

Из упомянутых работ были собраны данные значений метрик моделей на наборах данных. Не рассматривались метрики на обобщенных категориях задач (объединяющих только в рамках данной статьи несколько наборов данных по общей тематике), как, например, данные в таблице 7 статьи Open-FinLLMs. В работе использовалась min-max нормализация, и нормализованное значение метрики X составляло $\frac{X'-X_{min}}{X_{max}-X_{min}}$, где X' – первоначальное значение метрики, X_{max} и X_{min} – максимальное и минимальное значения метрики соответственно. Так как далее ко всем значениям применялась эта нормализация, то не рассматривались сеты, на которых считались метрики, не имеющие четкого диапазона возможных значений (так, например, не рассматривалась MSE). У наборов данных, имеющих

более одной подсчитанной метрики, брались те значения, которые соотносятся с метриками того же набора данных в других исследованиях. В отсутствие таковых использовались в первую очередь метрики с возможным диапазоном значений от 0 до 1.

В итоге была получена матрица, состоящая из 34 моделей и 42 наборов данных, с 531 известными значениями метрик (37.18% заполненности), представленная в таблицах, размещенных на внешнем ресурсе[1].

Анализ влияния разреженности. Сперва была протестирована устойчивость матричной факторизации к разреженности матрицы. Для этого на основе полностью заполненной таблицы из бенчмарка FLaME были сформированы 1000 урезанных таблиц (случайный выбор 80% столбцов и 80% строк). После чего из каждой новой матрицы случайным образом были выброшены от 10% до 90% значений (они использовались в качестве тестовых данных). В качестве гиперпараметров были взяты factors=35, learning rate = 0.05, regularization = 0.01, epochs = 200 (гиперпараметры, соотносящиеся с дальнейшими экспериментами). На рисунке 1 представлены 95% доверительные интервалы и средняя ошибка $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$, где n – размер выборки, y_i и $\hat{y}_i$ – реальные и предсказанные метрики.

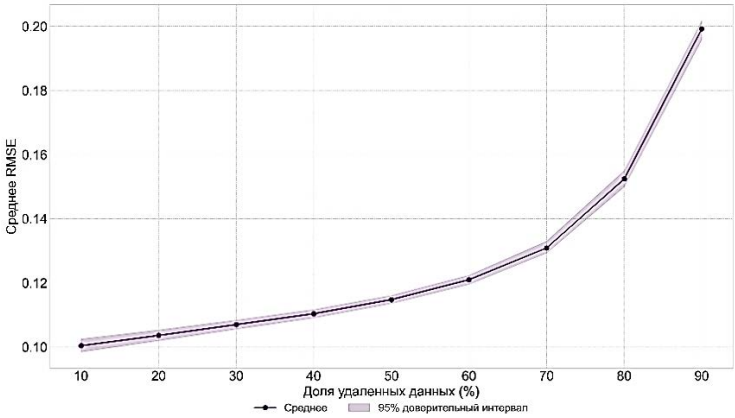


Рис. 1. RMSE в зависимости от процента пропущенных значений

[1] https://github.com/careepy/FinMF

Хорошо видно, что резкое ухудшение в качестве начинается в районе 70% пропущенных значений (30% заполненность матрицы). Разреженность составленной для экспериментов матрицы соотносится с данными результатами (37.18%).

Гиперпараметры. Эксперимент по заполнению пропущенных значений матрицы предсказанными метриками на основе матричной факторизации проводился с использованием пакета matrix-factorization[2]. Чтобы найти требуемые значения из уравнения 1, этот пакет использует метод стохастического градиентного спуска [50] и минимизирует уравнение 2 [16].

Изначально проводился поиск подходящих гиперпараметров, применяя поиск по сетке по широкому диапазону значений. Так, были рассмотрены factors: 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50; learning rate: 0.5, 0.1, 0.05, 0.01, 0.005, 0.001; regularization: 0.5, 0.1, 0.05, 0.01, 0.005, 0.001. Считалось среднее значение метрики RMSE по всем комбинациям с использованием пятикратной 5-блочной кросс-валидации (RepeatedKFold [51]). Результаты поиска показали, что наименьшая средняя ошибка достигается при значениях параметра learning rate 0.05, regularization 0.01 и числа скрытых факторов 20 (рисунок 2). Для данных параметров определялось оптимальное количество эпох — около 200, на основе графика динамики ошибки (рисунок 3). С использованием описанных выше гиперпараметров были получены итоговые предсказания R_{res}, которые представлены в таблицах, размещенных на внешнем ресурсе[3], в которых также приведены значения ошибок для показателей, которые были известны изначально.

Ошибки были рассчитаны как разница между фактическими и предсказанными измерениями. Помимо этого, также фиксируется значение $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}|$, (где n – размер выборки, y_i и $\widehat{y_i}$ – реальные и предсказанные метрики) итоговой модели (0.07), так как оно будет использовано при оценке агрегированной метрики. Стоит отметить, что, хотя в результирующей таблице этого не наблюдалось, такой подход может привести к тому, что предсказанные показатели превысят 1. Важно отметить, что этот подход позволяет найти только приблизительные показатели и показывает сложные, но относительные различия в результирующих оценках.

[2] https://github.com/Quang-Vinh/matrix-factorization
[3] https://github.com/careepy/FinMF

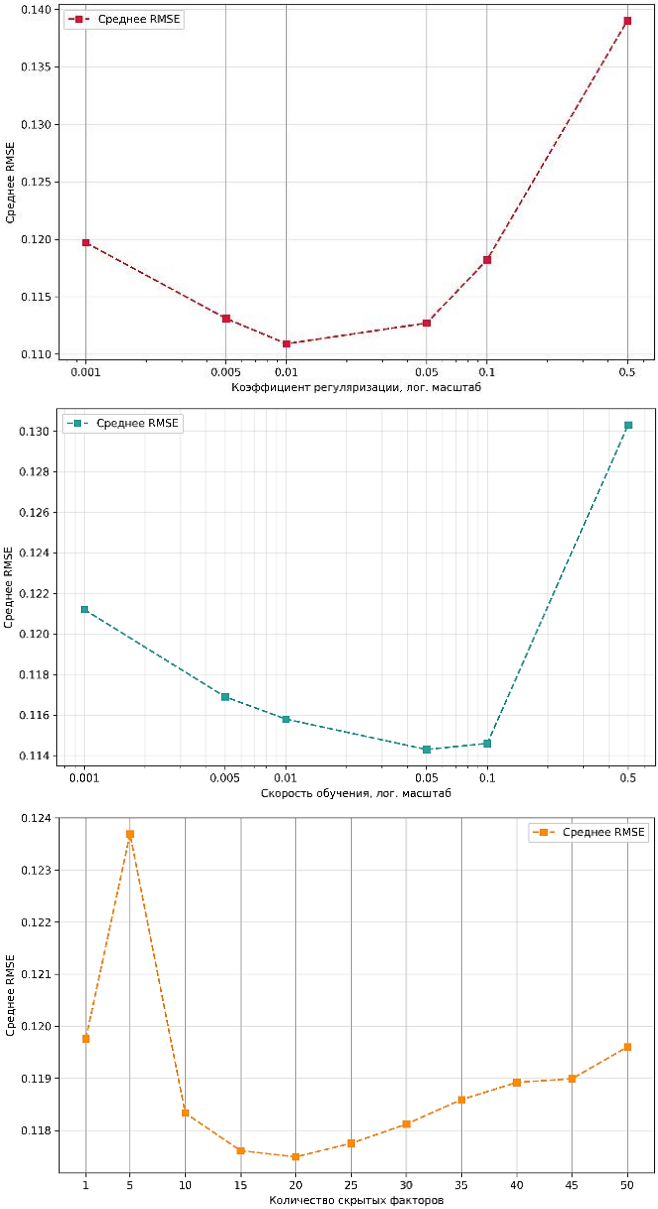


Рис. 2. Средняя RMSE для гиперпараметров регуляризации, скорости обучения и числа скрытых факторов

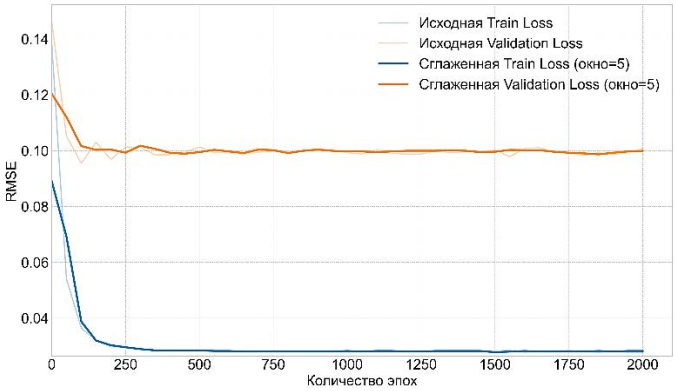


Рис. 3. Зависимость RMSE от количества эпох

Агрегированная метрика LLM. Составленная таблица с оценками значений метрик не позволит напрямую сравнивать и ранжировать модели. Для этого предлагается ввести агрегирующей метрику A, которая для каждой модели m суммирует нормализованные значения метрик по всем наборам данных:

$$A_m = \sum_{d \in D} A_m^d = \sum_{d \in D} \frac{d_m - d_{min}}{d_{max} - d_{min}},$$

где D – наборы данных, d – один выбранный набор данных, d_m – значение метрики модели на наборе данных d, d_{min} и d_{max} – минимальные и максимальные полученные по всем моделям значения метрик на наборе данных d. Итоговые отранжированные значения A_m, полученные по R_{res}, представлены в таблице 1.

Важно также рассмотреть, насколько значение метрики A зависит от ошибок предсказаний и оценить возможные значения метрики A при различных предположениях. В данной работе будет предложено два метода: с использованием вариации на ошибку MAE и с использованием метода Монте-Карло.

Первый метод заключается в том, что рассматриваются максимальные и минимальные значения A_m при вариации предсказаний на MAE, вычисленное при кросс-валидации (0.07). MAE было выбрано, потому что оно дает косвенную оценку потенциальной ошибки. Для этого:

– для каждого набора данных d вычисляются все возможные d_{min} и d_{max} при различных вариациях предсказаний;

– вычисляются максимальные и минимальные возможные A_m^d при различных d_{min} и d_{max} и различных вариациях d_m;

– верхняя граница A_m – сумма максимальных A_m^d, нижняя – сумма минимальных.

Часть полученных таким образом интервалов указана на рисунке 4, на котором представлены оценки для некоторых моделей, полученные при использовании метода Монте-Карло и MAE. По вертикали отложена предсказанная оценка A_m, по горизонтали – LLM (числовые значения указывают на положение в итоговом рейтинге).

Для подтверждения полученных выше результатов предложим еще один способ, использующий метод Монте-Карло. Он заключается в вычислении различных A_m при вариациях d_m на случайные значения из распределения ошибок:

– Оценивается распределение ошибок. Для этого генерируется N^{sp} разбиений на обучающие (80%) и тестовые (20%) выборки и на тестовых выборках вычисляются ошибки предсказаний.

– Генерируется N^{tb} таблиц, где каждая ячейка — сумма соответствующей ячейки R_{res} и случайной ошибки из распределения, определенного на шаге 1.

– Для каждой таблицы t из шага 2 и модели m вычисляется A_m^t.

– Для каждой m вычисляется среднее по A_M^* и определяется доверительный интервал (5 и 95 перцентили).

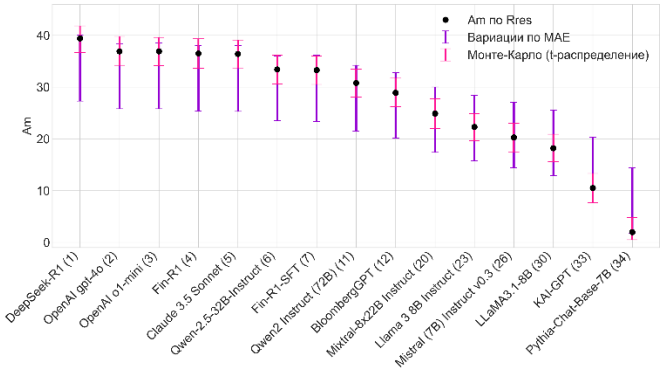


Рис. 4. Интервалы, построенные с помощью MAE и метода Монте Карло для некоторых моделей

Для расчета было взято $N^{tb} = N^{sp} = 1000$. На рисунке 5 представлено полученное эмпирическое распределение ошибок и аппроксимирующее его распределение Стьюдента со степенями свободы = 4.71 и матожиданием = –0.002. По оси X отложены значения ошибок, а по оси Y – плотность вероятности. Без учета выбросов стандартное отклонение составляет 0.063. Таким образом, случайные ошибки были взяты из распределения Стьюдента с озвученными параметрами. Итоговые A_m и доверительные интервалы для части моделей показаны на рисунке 4. Полные полученные значения можно найти в таблице 1.

Таблица 1. Значения метрики A и границ интервалов MAE и Монте-Карло

№	Модель	A	A_{min}^{MAE}	A_{max}^{MAE}	A_{min}^{MC}	A_{max}^{MC}
1	DeepSeek R1	39,9	27,6	40,5	37,1	42,7
2	OpenAI gpt-4o	37,0	25,6	38,5	34,3	39,8
3	OpenAI o1-mini	36,9	25,7	38,5	34,1	39,6
4	Fin-R1	36,6	25,4	38,2	33,7	39,5
5	Claude 3,5 Sonnet	36,5	25,2	38,1	33,7	39,1
6	Qwen-2,5-32B-Instruct	33,3	23,2	36,1	30,5	36,1
7	Fin-R1-SFT	33,2	23,2	36,0	30,3	36,0
8	DeepSeek-V3	33,1	22,9	35,7	30,5	35,7
9	Qwen-2,5-14B-Instruct	32,1	22,3	35,2	29,3	35,1
10	Llama 3 70B Instruct	31,3	21,8	34,6	28,5	34,1
11	Qwen 2 Instruct (72B)	30,9	21,4	34,3	28,1	33,5
12	Google Gemini 1,5 Pro	29,1	20,1	33,0	26,3	31,8
13	BloombergGPT	29,0	20,0	32,8	26,3	31,7
14	Gemma 2 27B	28,6	19,8	32,6	25,8	31,3
15	Gemma 2 9B	26,6	18,5	31,4	23,8	29,5
16	Jamba 1,5 Large	26,4	18,4	31,2	23,6	29,1
17	Claude 3 Haiku	26,3	18,4	31,2	23,5	29,0
18	Cohere Command R +	26,1	18,0	30,8	23,3	28,9
19	WizardLM-2 8x22B	25,1	17,4	30,2	22,3	27,8
20	Mixtral-8x22B Instruct	24,4	16,9	29,8	21,5	27,2
21	Qwen-2,5-7B-Instruct	24,1	17,0	29,8	21,3	27,0
22	Cohere Command R 7B	24,0	16,7	29,5	21,1	26,6
23	QwQ-32B-Preview	22,2	15,4	28,3	19,6	24,8
24	Llama 3 8B Instruct	21,9	15,3	28,2	19,2	24,5
25	Mixtral-8x7B Instruct	21,8	15,3	28,1	19,2	24,6
26	Mistral (7B) Instruct v0,3	20,0	14,0	26,8	17,0	22,7
27	DeepSeek LLM (67B)	19,6	13,7	26,5	16,7	22,3
28	FinLLaMA	18,7	13,2	26,0	15,7	21,4
29	Jamba 1,5 Mini	18,1	12,6	25,5	15,2	20,7
30	LLaMA3,1-8B	18,0	12,6	25,5	15,5	20,7
31	DBRX Instruct	12,9	9,0	21,9	10,2	15,6
32	LLaMA3-8B	11,5	8,1	21,0	8,8	14,3
33	KAI-GPT	10,2	7,3	20,1	7,4	12,9
34	Pythia-Chat-Base-7B	1,5	1,2	14,0	0,0	4,3

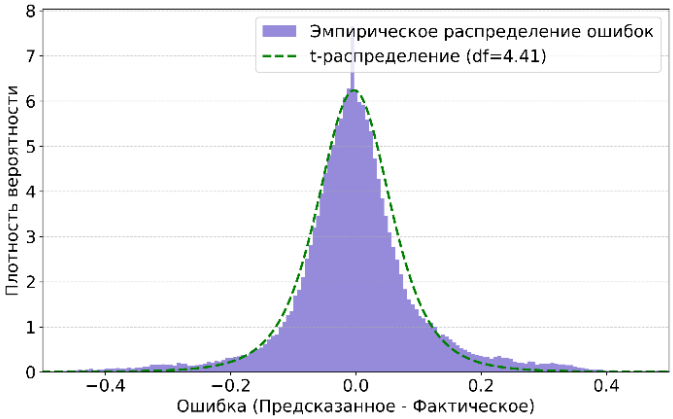


Рис. 5. Эмпирическое распределение ошибок и распределение Стьюдента

5. Обсуждение. Данный раздел посвящен обсуждению и интерпретации результатов, полученных в ходе эксперимента по применению матричной факторизации. Обсуждение разбито на две части: анализ итогового рейтинга моделей и анализ ошибок.

Анализ результатов. В данном подразделе представлен анализ результатов, полученных с использованием метода матричной факторизации для набора больших языковых моделей. Оценка качества на основе агрегированной метрики A, а также её оценок, полученных с помощью вариации по MAE и метода Монте-Карло.

Итоговый рейтинг моделей (таблица 1 и рисунок 4) показывает следующую тенденцию: оценка качества ведущих больших языковых моделей сблизилась до такой степени, что объявить однозначного лидера становится затруднительно. Разница в итоговых метриках между моделями из верхней части рейтинга, представленного в таблице 1 (DeepSeek R1, OpenAI gpt-4o, OpenAI o1-mini, Fin-R1, Claude 3.5 Sonnet), оказывается в пределах погрешности. Хоть DeepSeek R1 имеет небольшой отрыв от всех остальных моделей, существующий отрыв является минимальным. Так, если посмотреть на группу моделей, включающую OpenAI gpt-4o (36.96), OpenAI o1-mini (36.85), Fin-R1 (36.63) и Claude 3.5 Sonnet (36.47), то видно, что их агрегированные показатели A отличаются на десятые доли балла.

Ключевым аспектом является пересечение их доверительных интервалов, рассчитанных методом Монте-Карло (A^{MC}_{min} и A^{MC}_{max} в таблице 1). Например, диапазон оценки качества для gpt-4o

(34.27-39.85), o1-mini (34.10-39.59), Fin-R1 (33.74-39.48) и Claude 3.5 Sonnet (33.73-39.09) в значительной степени совпадает. Это говорит о том, что при повторных тестах или на несколько ином наборе данных их порядок в рейтинге мог бы легко измениться.

Однако здесь стоит особенно выделить модель Fin-R1, имеющую в своей архитектуре всего 7 миллиардов параметров, и находящуюся на первых строчках рейтинга, в то время как другие модели того же диапазона оценки качества имеют сотни миллионов параметров.

Невысокие результаты получили модели KAI-GPT и Pythia-Chat-Base-7B. В исходной таблице эти модели представляют собой особый случай в рамках данного исследования. Исходные метрики для них были предоставлены исключительно для двух наборов данных (KAI-GPT Eval. Set №1 и KAI-GPT Eval. Set №2), в которых не оценивалась ни одна другая модель из выборки. Несмотря на это ограничение, итоговый рейтинг, сгенерированный методом, последовательно размещает KAI-GPT и Pythia-Chat-Base-7B на низких позициях в общем зачете. Однако, предсказанные значения на остальных 30+ наборах данных скорее являются результатом обобщения модели на усредненных данных всей матрицы, нежели реальной оценкой их метрик.

Анализ ошибок предсказания матричной факторизации. Средняя абсолютная ошибка предсказания по всем известным значениям составляет 0.024 – данный показатель свидетельствует о том, что модель успешно справилась с описанием исходных данных через латентные представления. Средняя абсолютная ошибка для каждой отдельной модели также не превышает 0.033, по наборам данных – 0.04.

Дополнительно метод помогает выявить специфические особенности моделей на конкретных задачах, когда ошибки предсказания аномально высоки или низки. Так, ошибок со значением по модулю, большим 0.1, всего 5: Google Gemini 1,5 Pro – Banking77 (0.158), DeepSeek LLM – TAT-QA (0.151), Gemma 2 9B – ECTSum (0.129), DBRX Instruct – Federal Open Market Committee (0.113), QwQ-32B-Preview – REFinD (0.1).

Самая высокая полученная ошибка (0.158) принадлежит модели Google Gemini 1,5 Pro на наборе данных Banking77. Такой результат может объясняться тем, что в итоговом рейтинге Gemini 1,5 Pro находится на 12 месте, но в исходной матрице значение метрики для данной модели на наборе данных Banking77 ниже всех остальных на 0.09 пунктов. Это выявляет специфическую особенность модели (возможно, в многоклассовой классификации намерений в узкой

банковской тематике), которую не удалось уловить обобщенным латентным представлениям.

DeepSeek LLM продемонстрировал самые низкие метрики на TAT-QA (разница от 0.19), задаче, связанной с анализом табличных данных в финансовой области. Gemma 2 9B показала невысокие результаты на наборе данных ECTSum, связанном с суммаризацией длинных стенограмм, отчетов об убытках и прибылях компаний. Высокая метрика ошибки на DBRX Instruct, возможно, связана с тем, что модель в целом демонстрирует невысокие показатели на финансовых задачах, в особенности на наборе данных FOMC (на 0.2 ниже ближайшей модели). QwQ-32B-Preview демонстрирует результаты сильно ниже среднего на REFinD (0.2 пункта) – задаче, связанной с извлечением связей в финансовых и юридических документах.

Все полученные ошибки представлены в таблицах, размещенных на внешнем ресурсе[4].

6. Заключение. В данной работе была оценена применимость матричной факторизации для предсказания неизвестных метрик качества LLM на финансовых наборах данных. Для предсказанных значений была представлена агрегированная метрика, позволяющая ранжировать модели. Исследование, охватившее 42 набора данных и 34 модели, продемонстрировало потенциальную применимость использования матричной факторизации в качестве инструмента для предсказания оценки качества LLM. Предложенный метод может способствовать выбору моделей для решения финансовых задач.

Проведенный эксперимент показал, что оценка качества ведущих LLM, таких как DeepSeek R1, OpenAI gpt-4o, OpenAI o1-mini, Fin-R1 и Claude 3.5 Sonnet, находится приблизительно на одном уровне. Различия их итоговых показателей настолько малы, что попадают в пределы погрешности, и не позволяют выделить явного лидера. Данный вывод был подкреплен анализом доверительных интервалов, рассчитанных с помощью метода Монте-Карло.

Однако стоит подчеркнуть, что данный эксперимент показывает только общую эмпирическую картину. Некоторые значения метрик могут не отображать реальные метрики, полученные на новых наборах данных. Полученные ошибки могут быть вызваны разными использованными промптами, нюансами в тестовых данных, использованием разных языков, погрешностями факторизации. Анализ ошибок позволяет выявить чрезмерно хороший/плохой результат

[4] https://github.com/careepy/FinMF

на известных данных, но не позволяет предсказать подобное на новых данных.

Также, хоть матричная факторизация в целом подходит для решения задачи заполнения неизвестных значений метрик, следует проверять ее применимость при каждом новом эксперименте.

Дальнейшие исследования могут быть направлены на использование более сложных методов предсказания неизвестных значений метрик, в том числе нелинейных.

Литература

1. Ali C.-S.-M., Mahmood I. A Comprehensive Survey on Large Language Models: Architectures, Applications, and Ethical Considerations // Engineering and Technology Journal. 2025. vol. 10. no. 04. pp. 4578–4593. DOI: 10.47191/etj/v10i04.26.
2. Zhao H., Liu Z., Wu Z., Li Y., Yang T., Shu P., Xu S., Dai H., Zhao L., Mai G., Liu N., Liu T. Revolutionizing finance with LLMs: An overview of applications and insights // arXiv preprint arXiv:2401.11641. 2024.
3. Dhake S.-P., Lassi L., Hippalgaonkar A., Gaidhani R.-A., Jyothi N.-M. Impacts and Implications of Generative AI and Large Language Models: Redefining Banking Sector // Journal of Informatics Education and Research. 2024. vol. 4. no. 2. pp. 248–257. DOI: 10.52783/jier.v4i2.767.
4. Zhao W.-X., Liu J., Ren R., Wen J.-R. Dense text retrieval based on pretrained language models: A survey // ACM Transactions on Information Systems. 2024. vol. 42. no. 4. pp. 1–60. DOI: 10.1145/3637870.
5. Luo B., Zhang Z., Wang Q., Ke A., Lu S., He B. AI-powered fraud detection in decentralized finance: A project life cycle perspective // arXiv preprint arXiv:2308.15992. 2023.
6. Feng D., Dai Y., Huang J., Zhang Y., Xie Q., Han W., Lopez-Lira A., Wang H. Empowering many, biasing a few: Generalist credit scoring through large language models // arXiv preprint arXiv:2310.00566. 2023.
7. Dong Y., Yan D., Almudaifer A.-I., Yan S., Jiang Z., Zhou Y. BELT: A pipeline for stock price prediction using news. IEEE International Conference on Big Data // IEEE. 2020. pp. 1137–1146. DOI: 10.1109/BigData50022.2020.9378345.
8. Zhao W.-X., Zhou K., Li J., et al. A Survey of Large Language Models // arXiv preprint arXiv:2303.18223. 2023.
9. Li Y., Wang S., Ding H., Chen H. Large language models in finance: A survey // Proceedings of the Fourth ACM International Conference on AI in Finance (ICAIF '23). 2023. pp. 374–382. DOI: 10.1145/3604237.3626869.
10. Chen Z., Chen W., Smiley C., et al. FinQA: A Dataset of Numerical Reasoning over Financial Data // Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2021. pp. 3697–3711. DOI: 10.18653/v1/2021.emnlp-main.300.
11. Shah A., Gullapalli A., Vithani R., Galarnyk M., Chava S. FiNER-ORD: Financial Named Entity Recognition Open Research Dataset // arXiv preprint arXiv:2302.11157. 2023.
12. Tang Y., Yang Y., Huang A., Tam A., Tang J. FinEntity: Entity-level Sentiment Classification for Financial Texts // Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2023. pp. 15465–15471. DOI: 10.18653/v1/2023.emnlp-main.956.
13. Mukherjee R., Bohra A., Banerjee A., et al. ECTSum: A New Benchmark Dataset For Bullet Point Summarization of Long Earnings Call Transcripts // Proceedings of the

Conference on Empirical Methods in Natural Language Processing. 2022. pp. 10893–10906. DOI: 10.18653/v1/2022.emnlp-main.748.
14. Xie Q., Han W., Chen Z., et al. The FinBen: An Holistic Financial Benchmark for Large Language Models // arXiv preprint arXiv:2402.12659. 2024.
15. Chang Y., Wang X., Wang J., et al. A Survey on Evaluation of Large Language Models // ACM Transactions on Intelligent Systems and Technology. 2024. vol. 15. no. 3. pp. 1–45. DOI: 10.1145/3641289.
16. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems // Computer. 2009. vol. 42. no. 8. pp. 30–37. DOI: 10.1109/MC.2009.263.
17. Zhao Q., Xu M., Gupta K., et al. Can We Predict Performance of Large Models across Vision-Language Tasks // arXiv preprint arXiv:2410.10112. 2024.
18. Zhang Q., Lyu F., Liu X., Ma C. Collaborative Performance Prediction for Large Language Models // Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2024. pp. 2576–2596. DOI: 10.18653/v1/2024.emnlp-main.150.
19. Zhong X.-X., Yi C., Ye H.-J. Efficient Evaluation of Large Language Models via Collaborative Filtering // arXiv preprint arXiv:2504.08781. 2025.
20. Laskar M.-T.-R., Alqahtani S., Bari M.-S., et al. A Systematic Survey and Critical Review on Evaluating Large Language Models: Challenges, Limitations, and Recommendations // Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2024. pp. 13785–13816. DOI: 10.18653/v1/2024.emnlp-main.764.
21. Owen D. How predictable is language model benchmark performance // arXiv preprint arXiv:2401.04757. 2024.
22. Matlin G., Okamoto M., Pardawala H., Yang Y., Chava S. Finance Language Model Evaluation (FLaME) // Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics. 2025. pp. 880–926. DOI: 10.48550/arXiv.2506.15846.
23. Huang J., Xiao M., Li D., et al. Open-FinLLMs: Open Multimodal Large Language Models for Financial Applications // arXiv preprint arXiv:2408.11878. 2024.
24. Liu Z., Guo X., Lou F., et al. Fin-R1: A Large Language Model for Financial Reasoning through Reinforcement Learning // arXiv preprint arXiv:2503.16252. 2025.
25. KAI-GPT: The First Large Language Model Purpose-Built for Banking // Kasisto. URL: https//kasisto.com/blog/kai-gpt-the-first-large-language-model-purpose-built-for-banking/ (дата обращения: 25.11.2025).
26. Qian L., Zhou W., Wang Y., Peng X., Huang J., Xie Q. Fino1: On the Transferability of Reasoning-Enhanced LLMs and Reinforcement Learning to Finance // arXiv preprint arXiv:2502.08127. 2025.
27. Sharma S., Nayak T., Bose A., et al. FinRED: A Dataset for Relation Extraction in Financial Domain // Companion Proceedings of the Web Conference (WWW '22). 2022. pp. 595–597. DOI: 10.1145/3487553.3524637.
28. Kaur S., Smiley C., Gupta A., et al. REFinD: Relation Extraction Financial Dataset // Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023. pp. 3054–3063. DOI: 10.1145/3539618.3591911.
29. Sharma S., Khatuya S., Hegde M., et al. Financial Numeric Extreme Labelling: A dataset and benchmarking // Findings of the Association for Computational Linguistics: ACL 2023. 2023. pp. 2933–2946. DOI: 10.18653/v1/2023.findings-acl.219.
30. Sinha A., Khandait T. Impact of News on the Commodity Market: Dataset and Results // Advances in Information Retrieval (ECIR 2021). 2021. pp. 589–601. DOI: 10.1007/978-3-030-73103-8_41.
31. Yang L., Kenny E., Ng T.-L., Yang Y., Smyth B., Dong R. Generating Plausible Counterfactual Explanations for Deep Transformers in Financial Text Classification // Proceedings of the 28th International Conference on Computational Linguistics. 2020. pp. 6150–6160. DOI: 10.18653/v1/2020.coling-main.541.

32. Pardawala H., Sukhani S., Shah A., et al. SubjECTive-QA: Measuring Subjectivity in Earnings Call Transcripts' QA Through Six-Dimensional Feature Analysis // Proceedings of the 38th International Conference on Neural Information Processing Systems (NIPS '24). 2024. pp. 59342–59372.
33. Zhu F., Lei W., Chao Y., et al. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics. 2021. pp. 3277–3287. DOI: 10.18653/v1/2021.acl-long.254.
34. Chen Z., Li S., Smiley C., et al. ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering // Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2022. pp. 6279–6290. DOI: 10.18653/v1/2022.emnlp-main.421.
35. Malo P., Sinha A., Korhonen P., Wallenius J., Takala P. Good debt or bad debt: Detecting semantic orientations in economic texts // Journal of the Association for Information Science and Technology. 2014. vol. 65. no. 4. pp. 782–796. DOI: 10.1002/asi.23062.
36. Cortis K., Freitas A., Daudert T., et al. SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News Headlines // Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). 2017. pp. 519–535. DOI: 10.18653/v1/S17-2089.
37. Twitter Financial News Sentiment // Hugging Face: Zeroshot. 2024. URL: https//huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment (дата обращения: 25.11.2025).
38. Casanueva I., Temcinas T., Gerz D., Henderson M., Vulic I. Efficient Intent Detection with Dual Sentence Encoders // Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI. 2020. pp. 38–45. DOI: 10.18653/v1/2020.nlp4convai-1.5.
39. Mariko D., Abi-Akl H., Labidurie E., et al. The Financial Document Causality Detection Shared Task (FinCausal 2020) // Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation. 2020. pp. 23–32. DOI:10.48550/arXiv.2012.02505
40. Shah A., Hiray A., Shah P., et al. Numerical Claim Detection in Finance: A New Financial Dataset, Weak-Supervision Model, and Market Analysis // Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER). 2024. pp. 170–185. DOI: 10.18653/v1/2024.fever-1.21.
41. Chen C.-C., Lin C.-Y., Chiu C.-J., et al. Overview of the NTCIR-17 FinArg-1 Task: Fine-grained argument understanding in financial analysis // Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies. 2023. pp. 16–20. DOI: 10.20736/0002001323.
42. Zhao Y., Liu H., Long Y., et al. FinanceMATH: Knowledge-Intensive Math Reasoning in Finance Domains // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024. vol. 1. pp. 12841–12858. DOI: 10.18653/v1/2024.acl-long.693.
43. Zhao Y., Long Y., Liu H., et al. DocMath-Eval: Evaluating Math Reasoning Capabilities of LLMs in Understanding Long and Specialized Documents // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024. vol. 1. pp. 16103–16120. DOI: 10.18653/v1/2024.acl-long.852.
44. Yin Y., Yang Y., Yang J., Liu Q. FinPT: Financial Risk Prediction with Profile Tuning on Pretrained Foundation Models // arXiv preprint arXiv:2308.00065. 2024.
45. Hofmann H. Statlog (German Credit Data) // UCI Machine Learning Repository. 1994. URL: https//archive.ics.uci.edu/dataset/144/statlog+german+credit+data (дата обращения: 25.11.2025).

46. Quinlan R. Statlog (Australian Credit Approval) // UCI Machine Learning Repository. URL: https//archive.ics.uci.edu/dataset/143/statlog+australian+credit+approval (дата обращения: 25.11.2025).
47. Flowers J.G. Finance Instruct 500k // Hugging Face. 2025. URL: https//huggingface.co/datasets/Josephgflowers/Finance-Instruct-500k (дата обращения: 25.11.2025).
48. Financial Evaluation Dataset // GitHub: Alipay Team. 2023. URL: https//github.com/alipay/financial_evaluation_dataset (дата обращения: 25.11.2025).
49. Shah A., Paturi S., Chava S. Trillion Dollar Words: A New Financial Dataset, Task & Market Analysis // Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. 2023. vol. 1. pp. 6664–6679. DOI: 10.18653/v1/2023.acl-long.368.
50. Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent // Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT '2010). Physica-Verlag HD. 2010. pp. 177–186. DOI: 10.1007/978-3-7908-2604-3_16.
51. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection // Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95). 1995. vol. 2. pp. 1137–1143.

Вяткин Артем Андреевич — младший научный сотрудник, лаборатория прикладного искусственного интеллекта, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук (СПб ФИЦ РАН). Область научных интересов: вероятностные графические модели, алгебраические байесовские сети, нечеткие вычисления, большие языковые модели. Число научных публикаций — 24. aav@dscs.pro; 14-линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(812)508-3311.

Попцов Александр Владимирович — стажер-исследователь, лаборатория прикладного искусственного интеллекта, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук (СПб ФИЦ РАН). Область научных интересов: обработка естественного языка, большие языковые модели, ИИ-агенты. Число научных публикаций — 2. avp@dscs.pro; 14-линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(927)887-1617.

Олисеенко Валерий Дмитриевич — научный сотрудник, лаборатория прикладного искусственного интеллекта, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук (СПб ФИЦ РАН). Область научных интересов: обработка естественных языков, глубокое обучение, анализ социальных сетей, информационная безопасность, социоинженерные атаки. Число научных публикаций — 60. vdo@dscs.pro; 14-я линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(812)508-3311.

Абрамов Максим Викторович — канд. техн. наук, доцент, руководитель лаборатории, лаборатория прикладного искусственного интеллекта, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук (СПб ФИЦ РАН). Область научных интересов: генеративный искусственный интеллект, большие языковые модели, машинное обучение, анализ данных, социоинженерные атаки. Число научных публикаций — 170. mva@dscs.pro; 14-линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(812)508-3311.

Поддержка исследований. Работа выполнена в рамках проекта по государственному заданию СПб ФИЦ РАН № FFZF-2024-0003.

DOI 10.15622/ia.25.2.1

A. VYATKIN, A. POPTSOV, V. OLISEENKO, M. ABRAMOV

RESEARCH ON THE APPLICABILITY OF MATRIX FACTORIZATION FOR RANKING LARGE LANGUAGE MODELS

Vyatkin A., Poptsov A., Oliseenko V., Abramov M. **Research on the Applicability of Matrix Factorization for Ranking Large Language Models.**

Abstract. In recent years, Large Language Models (LLMs) have gained widespread adoption in the financial domain. Direct comparison of models can be challenging, as datasets and LLMs may be closed, and evaluation parameters may vary. This paper proposes using the matrix factorization method from recommender systems, originally designed to predict user preferences, to address the task of predicting unknown metrics. The aim is to evaluate the applicability of matrix factorization for predicting LLM performance metrics on financial tasks, as well as to develop an LLM ranking method based on metric aggregation. An experiment involving the application of matrix factorization is conducted using data collected from academic research, covering 34 LLMs and 42 financial datasets. The average Mean Absolute Error (MAE) of the method across all runs is 0.07 on the test dataset. The top positions in the ranking are held by DeepSeek-R1, OpenAI GPT-4o, OpenAI o1-mini, Fin-R1, and Claude 3.5 Sonnet. The impact of prediction error on the final results is investigated using two approaches: analysis of MAE and the Monte Carlo method. The results are analyzed, yielding the following main conclusions: a) matrix factorization can be applied to predict missing model metric values on datasets; b) leading large language models have converged in performance to such an extent that identifying a clear leader is difficult; c) large prediction errors allow for the identification of specific model features on particular tasks. The proposed method can simplify the selection of a suitable model for financial tasks.

Keywords: large language models, performance evaluation, matrix factorization, financial domain.

References

1. Ali C.-S.-M., Mahmood I. A Comprehensive Survey on Large Language Models: Architectures, Applications, and Ethical Considerations. Engineering and Technology Journal. 2025. vol. 10. no. 04. pp. 4578–4593. DOI: 10.47191/etj/v10i04.26.
2. Zhao H., Liu Z., Wu Z., Li Y., Yang T., Shu P., Xu S., Dai H., Zhao L., Mai G., Liu N., Liu T. Revolutionizing finance with LLMs: An overview of applications and insights. arXiv preprint arXiv:2401.11641. 2024.
3. Dhake S.-P., Lassi L., Hippalgaonkar A., Gaidhani R.-A., Jyothi N.-M. Impacts and Implications of Generative AI and Large Language Models: Redefining Banking Sector. Journal of Informatics Education and Research. 2024. vol. 4. no. 2. pp. 248–257. DOI: 10.52783/jier.v4i2.767.
4. Zhao W.-X., Liu J., Ren R., Wen J.-R. Dense text retrieval based on pretrained language models: A survey. ACM Transactions on Information Systems. 2024. vol. 42. no. 4. pp. 1–60. DOI: 10.1145/3637870.
5. Luo B., Zhang Z., Wang Q., Ke A., Lu S., He B. AI-powered fraud detection in decentralized finance: A project life cycle perspective. arXiv preprint arXiv:2308.15992. 2023.
6. Feng D., Dai Y., Huang J., Zhang Y., Xie Q., Han W., Lopez-Lira A., Wang H. Empowering many, biasing a few: Generalist credit scoring through large language models. arXiv preprint arXiv:2310.00566. 2023.

7. Dong Y., Yan D., Almudaifer A.-I., Yan S., Jiang Z., Zhou Y. BELT: A pipeline for stock price prediction using news. IEEE International Conference on Big Data. IEEE. 2020. pp. 1137–1146. DOI: 10.1109/BigData50022.2020.9378345.
8. Zhao W.-X., Zhou K., Li J., et al. A Survey of Large Language Models. arXiv preprint arXiv:2303.18223. 2023.
9. Li Y., Wang S., Ding H., Chen H. Large language models in finance: A survey. Proceedings of the Fourth ACM International Conference on AI in Finance (ICAIF '23). 2023. pp. 374–382. DOI: 10.1145/3604237.3626869.
10. Chen Z., Chen W., Smiley C., et al. FinQA: A Dataset of Numerical Reasoning over Financial Data. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2021. pp. 3697–3711. DOI: 10.18653/v1/2021.emnlp-main.300.
11. Shah A., Gullapalli A., Vithani R., Galarnyk M., Chava S. FiNER-ORD: Financial Named Entity Recognition Open Research Dataset. arXiv preprint arXiv:2302.11157. 2023.
12. Tang Y., Yang Y., Huang A., Tam A., Tang J. FinEntity: Entity-level Sentiment Classification for Financial Texts. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2023. pp. 15465–15471. DOI: 10.18653/v1/2023.emnlp-main.956.
13. Mukherjee R., Bohra A., Banerjee A., et al. ECTSum: A New Benchmark Dataset For Bullet Point Summarization of Long Earnings Call Transcripts. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2022. pp. 10893–10906. DOI: 10.18653/v1/2022.emnlp-main.748.
14. Xie Q., Han W., Chen Z., et al. The FinBen: An Holistic Financial Benchmark for Large Language Models. arXiv preprint arXiv:2402.12659. 2024.
15. Chang Y., Wang X., Wang J., et al. A Survey on Evaluation of Large Language Models. ACM Transactions on Intelligent Systems and Technology. 2024. vol. 15. no. 3. pp. 1–45. DOI: 10.1145/3641289.
16. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. Computer. 2009. vol. 42. no. 8. pp. 30–37. DOI: 10.1109/MC.2009.263.
17. Zhao Q., Xu M., Gupta K., et al. Can We Predict Performance of Large Models across Vision-Language Tasks. arXiv preprint arXiv:2410.10112. 2024.
18. Zhang Q., Lyu F., Liu X., Ma C. Collaborative Performance Prediction for Large Language Models. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2024. pp. 2576–2596. DOI: 10.18653/v1/2024.emnlp-main.150.
19. Zhong X.-X., Yi C., Ye H.-J. Efficient Evaluation of Large Language Models via Collaborative Filtering. arXiv preprint arXiv:2504.08781. 2025.
20. Laskar M.-T.-R., Alqahtani S., Bari M.-S., et al. A Systematic Survey and Critical Review on Evaluating Large Language Models: Challenges, Limitations, and Recommendations. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2024. pp. 13785–13816. DOI: 10.18653/v1/2024.emnlp-main.764.
21. Owen D. How predictable is language model benchmark performance. arXiv preprint arXiv:2401.04757. 2024.
22. Matlin G., Okamoto M., Pardawala H., Yang Y., Chava S. Finance Language Model Evaluation (FLaME). Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics. 2025. pp. 880–926. DOI: 10.48550/arXiv.2506.15846.
23. Huang J., Xiao M., Li D., et al. Open-FinLLMs: Open Multimodal Large Language Models for Financial Applications. arXiv preprint arXiv:2408.11878. 2024.
24. Liu Z., Guo X., Lou F., et al. Fin-R1: A Large Language Model for Financial Reasoning through Reinforcement Learning. arXiv preprint arXiv:2503.16252. 2025.

25. KAI-GPT: The First Large Language Model Purpose-Built for Banking. Kasisto. Available at: https.kasisto.com/blog/kai-gpt-the-first-large-language-model-purpose-built-for-banking/ (accessed 25.11.2025).
26. Qian L., Zhou W., Wang Y., Peng X., Huang J., Xie Q. Fino1: On the Transferability of Reasoning-Enhanced LLMs and Reinforcement Learning to Finance. arXiv preprint arXiv:2502.08127. 2025.
27. Sharma S., Nayak T., Bose A., et al. FinRED: A Dataset for Relation Extraction in Financial Domain. Companion Proceedings of the Web Conference (WWW '22). 2022. pp. 595–597. DOI: 10.1145/3487553.3524637.
28. Kaur S., Smiley C., Gupta A., et al. REFinD: Relation Extraction Financial Dataset. Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023. pp. 3054–3063. DOI: 10.1145/3539618.3591911.
29. Sharma S., Khatuya S., Hegde M., et al. Financial Numeric Extreme Labelling: A dataset and benchmarking. Findings of the Association for Computational Linguistics: ACL 2023. 2023. pp. 2933–2946. DOI: 10.18653/v1/2023.findings-acl.219.
30. Sinha A., Khandait T. Impact of News on the Commodity Market: Dataset and Results. Advances in Information Retrieval (ECIR 2021). 2021. pp. 589–601. DOI: 10.1007/978-3-030-73103-8_41.
31. Yang L., Kenny E., Ng T.-L., Yang Y., Smyth B., Dong R. Generating Plausible Counterfactual Explanations for Deep Transformers in Financial Text Classification. Proceedings of the 28th International Conference on Computational Linguistics. 2020. pp. 6150–6160. DOI: 10.18653/v1/2020.coling-main.541.
32. Pardawala H., Sukhani S., Shah A., et al. SubjECTive-QA: Measuring Subjectivity in Earnings Call Transcripts' QA Through Six-Dimensional Feature Analysis. Proceedings of the 38th International Conference on Neural Information Processing Systems (NIPS '24). 2024. pp. 59342–59372.
33. Zhu F., Lei W., Chao Y., et al. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics. 2021. pp. 3277–3287. DOI: 10.18653/v1/2021.acl-long.254.
34. Chen Z., Li S., Smiley C., et al. ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2022. pp. 6279–6290. DOI: 10.18653/v1/2022.emnlp-main.421.
35. Malo P., Sinha A., Korhonen P., Wallenius J., Takala P. Good debt or bad debt: Detecting semantic orientations in economic texts. Journal of the Association for Information Science and Technology. 2014. vol. 65. no. 4. pp. 782–796. DOI: 10.1002/asi.23062.
36. Cortis K., Freitas A., Daudert T., et al. SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News Headlines. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). 2017. pp. 519–535. DOI: 10.18653/v1/S17-2089.
37. Twitter Financial News Sentiment. Hugging Face: Zeroshot. 2024. Available at: https.huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment (accessed 25.11.2025).
38. Casanueva I., Temcinas T., Gerz D., Henderson M., Vulic I. Efficient Intent Detection with Dual Sentence Encoders. Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI. 2020. pp. 38–45. DOI: 10.18653/v1/2020.nlp4convai-1.5.

39. Mariko D., Abi-Akl H., Labidurie E., et al. The Financial Document Causality Detection Shared Task (FinCausal 2020). Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation. 2020. pp. 23–32. DOI:10.48550/arXiv.2012.02505
40. Shah A., Hiray A., Shah P., et al. Numerical Claim Detection in Finance: A New Financial Dataset, Weak-Supervision Model, and Market Analysis. Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER). 2024. pp. 170–185. DOI: 10.18653/v1/2024.fever-1.21.
41. Chen C.-C., Lin C.-Y., Chiu C.-J., et al. Overview of the NTCIR-17 FinArg-1 Task: Fine-grained argument understanding in financial analysis. Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies. 2023. pp. 16–20. DOI: 10.20736/0002001323.
42. Zhao Y., Liu H., Long Y., et al. FinanceMATH: Knowledge-Intensive Math Reasoning in Finance Domains. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024. vol. 1. pp. 12841–12858. DOI: 10.18653/v1/2024.acl-long.693.
43. Zhao Y., Long Y., Liu H., et al. DocMath-Eval: Evaluating Math Reasoning Capabilities of LLMs in Understanding Long and Specialized Documents. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024. vol. 1. pp. 16103–16120. DOI: 10.18653/v1/2024.acl-long.852.
44. Yin Y., Yang Y., Yang J., Liu Q. FinPT: Financial Risk Prediction with Profile Tuning on Pretrained Foundation Models. arXiv preprint arXiv:2308.00065. 2024.
45. Hofmann H. Statlog (German Credit Data). UCI Machine Learning Repository. 1994. Available at: https.archive.ics.uci.edu/dataset/144/statlog+german+credit+data (accessed 25.11.2025).
46. Quinlan R. Statlog (Australian Credit Approval). UCI Machine Learning Repository. Available at: https.archive.ics.uci.edu/dataset/143/statlog+australian+credit+approval (accessed 25.11.2025).
47. Flowers J.G. Finance Instruct 500k. Hugging Face. 2025. Available at: https.huggingface.co/datasets/Josephgflowers/Finance-Instruct-500k (accessed 25.11.2025).
48. Financial Evaluation Dataset. GitHub: Alipay Team. 2023. Available at: https.github.com/alipay/financial_evaluation_dataset (accessed 25.11.2025).
49. Shah A., Paturi S., Chava S. Trillion Dollar Words: A New Financial Dataset, Task & Market Analysis. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. 2023. vol. 1. pp. 6664–6679. DOI: 10.18653/v1/2023.acl-long.368.
50. Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent. Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT '2010). Physica-Verlag HD. 2010. pp. 177–186. DOI: 10.1007/978-3-7908-2604-3_16.
51. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95). 1995. vol. 2. pp. 1137–1143.

Vyatkin Artyom — Junior researcher, Laboratory of applied artificial intelligence, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: probabilistic graph models, algebraic Bayesian networks, fuzzy computing, large language models, multi-agent systems. The number of publications — 24. aav@dscs.pro; 39, 14-th Line V.O., 199178, St. Petersburg, Russia; office phone: +7(812)508-3311.

Poptsov Alexander — Intern researcher, Laboratory of applied artificial intelligence, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: natural language processing, large language models, AI agents. The number of publications — 2. avp@dscs.pro; 39, 14-th Line V.O., 199178, St. Petersburg, Russia; office phone: +7(927)887-1617.

Oliseenko Valerii — Researcher, Laboratory of applied artificial intelligence, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: natural language processing, deep learning, social network analysis, information security, social engineering attacks. The number of publications — 60. vdo@dscs.pro; 39, 14-th Line V.O., 199178, St. Petersburg, Russia; office phone: +7(812)508-3311.

Abramov Maxim — Ph.D., Associate Professor, Head of laboratory, Laboratory of applied artificial intelligence, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: generative artificial intelligence, large language models, machine learning, data analysis, social engineering attacks. The number of publications — 170. mva@dscs.pro; 39, 14-th Line V.O., 199178, St. Petersburg, Russia; office phone: +7(812)508-3311.

Acknowledgements. This work was performed under the State Assignment of the St. Petersburg Federal Research Center of the Russian Academy of Sciences № FFZF-2024-0003.

УДК 004.032.26; 621.396 **DOI** 10.15622/ia.25.2.2

А.А. Толстых, А.Н. Голубинский

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ НА БАЗЕ ГЛУБОКОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ ПРОГНОЗА РАБОЧИХ ЧАСТОТ И ПОЛОС В СИСТЕМЕ КОГНИТИВНОГО РАДИО

Толстых А.А., Голубинский А.Н. **Интеллектуальный анализ данных на базе глубокого обучения с подкреплением для прогноза рабочих частот и полос в системе когнитивного радио.**

Аннотация. В работе предложен метод решения задачи выбора канала связи в когнитивном радио на основе информации о текущем состоянии всех доступных каналов связи с использованием математического аппарата обучения с подкреплением. Метод заключается в формализации задачи выбора каналов связи в терминах «среда-агент» и обучении агентов с помощью алгоритмов Reinforce, SARSA и A2C. Приведён расчёт затрат памяти на решение задачи выбора каналов связи классическими методами. Оценка по памяти составляет $\mathbf{4{\times}2^{2n}}$ байт для случайного состояния каналов (занят/свободен) и $\mathbf{4{\times}}n^2$ байт – для одного свободного канала на каждом шаге при решении задачи табличным алгоритмом Q-обучения. Приведены две различные формализации вознаграждения для агента в рамках решаемой задачи при использовании обучения с подкреплением – для тривиального случая (бинарная доступность / недоступность частотного канала) и для более сложного случая – с учётом мощности (в дБ) в выбранном канале связи. Ограничение на первую формализацию состоит в том, что на каждой итерации должен быть только один свободный канал связи из всех доступных. Вторая предложенная формализация функции вознаграждения не накладывает подобных ограничений и более универсальна. Проведены вычислительные эксперименты для обеих формализаций функции вознаграждения, агенты обучающиеся с помощью алгоритмов SARSA и A2C, в среднем, достигают безошибочного решения задачи за 8000 эпизодов обучения для обеих формализаций обучения в модельной задаче для различных реализаций агентов. Алгоритм REINFORCE не позволяет достигать безошибочного решения, однако, формализация вознаграждения с учётом мощности повышает стабильность обучения алгоритмом REINFORCE. Даны теоретические оценки вычислительной сложности рассматриваемых методов, согласующиеся с вычислительными экспериментами.

Ключевые слова: когнитивное радио, обучение с подкреплением, глубокое обучение, искусственная нейронная сеть, многослойный персептрон, функция вознаграждения, программно-определяемое радио, синтетические данные, аугментация, искусственный интеллект.

1. Введение. Функционирование когнитивных систем радиосвязи основывается на мониторинге радиочастотного спектра с целью идентификации и дальнейшего использования свободных полос частот. Классические подходы не всегда позволяют получить приемлемый на практике результат, в связи с этим, одним из возможных путей решения является применение технологий искусственного интеллекта [1 – 4], и, в частности, обучения с подкреплением [2, 5 – 14]. Таким образом, современный этап научно-технического развития

характеризуется переходом от автоматизации к интеллектуализации управления, при котором окончательное решение принимается интеллектуальной системой, на основании разнородных данных значительного объёма.

Следует отметить, что система когнитивного радио (CRS) – это радиосистема, использующая технологию, которая позволяет этой системе получать знания о своей среде эксплуатации и географической среде, об установившихся правилах и о своём внутреннем состоянии, динамически и автономно корректировать свои эксплуатационные параметры и протоколы согласно полученным знаниям для достижения заранее поставленных целей, обучаясь на основе полученных результатов [15].

Современное развитие аппаратных средств, программного обеспечения и баз данных, для эффективного нейросетевого моделирования на базе глубоких искусственных нейронных сетей, позволяет на сегодняшний день выделить актуальное для теории и практики научно-техническое направление – применение глубокого обучения с подкреплением для задач когнитивного радио [16 – 18].

Применение нейросетевого предиктора на основе глубокого обучения с подкреплением, позволяет проводить машинное обучение без разметки данных при использовании функции вознаграждения [19, 20], то есть при отсутствии «учителя».

Важным компонентом при этом является подготовка обучающих, валидационных и тестовых выборок, включающих композицию из реальных (например, сигналы, поступающие от трансивера на базе программно-определяемого радио или SDR- радиоприёмника) и аугментацию в виде синтетических данных для реализации нейросетевых предикторов (с формированием соответствующей базы данных для обучения).

Необходимо подчеркнуть, что отдельным приоритетным вопросом при решении поставленной задачи когнитивного радио является разработка и исследование адекватного критерия эффективности функционирования системы (например, состоящей из совокупности средств радиосвязи для различных режимов работы).

Цель работы – разработка метода прогноза рабочих частот и полос для системы когнитивного радио на базе глубокого обучения с подкреплением.

2. Постановка задачи. Рассмотрим радиоканал, в котором рабочий частотный диапазон разбит на полосы, и каждая полоса может быть занята или свободна для радиообмена данными. Математически это можно формализовать, сравнивая текущее значение мощности

(PdB_k) помехи в полосе с некоторым порогом (β), получив бинарное состояние: «1» – частотный канал (k) свободен, «0» – частотный канал занят:

$$\mathrm{PN}_k = \begin{cases} 1, \text{если } \mathrm{PdB}_k \le \beta; \\ 0, \text{иначе,} \end{cases} \quad (1)$$

где порог выбирается из состояния помехово-шумовой обстановки (например, β = -80дБм).

Требуется определить рабочие частоты и наибольшие полосы частот для динамически меняющейся электромагнитной обстановки (например, в которой присутствуют периодические и непериодические помехи, фон и другие шумы различной природы).

3. Описание метода. Определим в качестве критерия для решения поставленной задачи – максимизацию целевой функции в виде некоторого показателя качества прогноза. В этой связи необходимо максимизировать качество прогноза рекомендуемых свободных каналов:

$$\max[g_t], \quad (2)$$

где g_t – показатель качества прогноза, который определяется как функция от состояния радиоканалов в следующий момент времени. Например, как скалярное произведение соответствующих векторов и выглядит следующим образом:

$$g_t = a_t^T \mathrm{PN}_{t+1}, \quad (3)$$

здесь a_t – вектор-столбец действия агента в текущий момент времени («1» – канал рекомендуется использовать, «0» – канал не рекомендуется использовать); PN_{t+1} – вектор-столбец состояния радиоканалов на основе измеренной или моделируемой помехово-шумовой обстановки в следующий момент времени с элементами $\{\mathrm{PN}_k\}$ («1» – канал свободен, «0» – канал занят); «T» – операция транспонирования.

Функцию вознаграждения (r_{t+1}) можно определить, через функциональную зависимость от показателя качества прогноза или в первом приближении, возможно, использовать их равенство:

$$r_{t+1} = g_t. \quad (4)$$

Два основных подхода к обучению с подкреплением для безмодельных методов (не задействуют динамику переходов среды в явном виде) – алгоритмы, основанные на полезности и алгоритмы на базе стратегии (политики) [19, 20].

Например, если для решения задачи воспользоваться Q-обучением (основано на полезности), то на каждом шаге на основе ε-жадной стратегии, текущего состояния среды (s_t) и Q-матрицы (Q_t) формируется текущее действие агента a_t, на базе которого получается состояние среды, т.е. матрица прогноза состояния радиоканала («1» – канал свободен, «0» – канал занят) в следующий момент времени s_{t+1}, например:

$$s_{t+1} = a_t, \tag{5}$$

которая также может быть дополнена априорной информацией, помимо вектора-столбца a_t, векторами-столбцами о текущем и предыдущих состояниях радиоканала (например: PN_{t+1}, PN_t, PN_{t-1},…). Далее рассчитывается значение функции вознаграждения в следующий момент времени (r_{t+1}) и на его основе вычисляется Q-матрица значений ценности состояний в следующий момент времени (Q_{t+1}), используя такие гиперпараметры как скорость обучения (α) и коэффициент дисконтирования (γ) [19]. Настройка Q-функции осуществляется с помощью метода временных различий (TD-обучения, сочетающего в себе идеи метода Монте-Карло и динамического программирования) и использования итерационного метода для решения уравнения оптимальности Беллмана [20].

Существенным ограничением для непосредственного использования алгоритмов табличных методов решения на основе полезности и на основе стратегии является значительное множество возможных состояний (для Q-обучения – это большая размерность Q-матрицы, а для обучения на базе стратегии – большая размерность π-матрицы вероятностей действия в соответствующем состоянии). Например, при n одновременно анализируемых частотных каналов при случайном состоянии канала («свободен»/«занят») количество состояний среды (N) определяется выражением (количество строк Q-матрицы):

$$N = 2^n, \tag{6}$$

а количество возможных действий агента (M) также рассчитывается по формуле (количество столбцов Q-матрицы):

$$M = 2^n. \quad (7)$$

В результате Q-матрица содержит N×M элементов:

$$N \times M = 2^{2n}, \quad (8)$$

и для её описания при одинарной точности (FP32) потребуется:

$$V = 4 \times 2^{2n} \text{ байт}. \quad (9)$$

При n=16 следует, что только для хранения Q-матрицы потребуется 16 ГБ, а при n=20 необходимо 4 ТБ.

Однако, если рассмотреть тривиальный частный случай, например, когда из n каналов всегда свободен только один, то размер Q-матрицы существенно уменьшается, а расчётные формулы принимают вид:

$$N = n; \quad M = n, \quad N \times M = n \times n, \quad V = 4\,n^2 \text{ байт}, \quad (10)$$

тогда при n=16 следует, что для хранения Q-матрицы потребуется 1 КБ, а при n=20 необходимо 1,6 КБ.

Таким образом, наличие закономерностей (например, детерминированных) появления помех в радиоканалах позволяет определённым образом уменьшить число возможных состояний, однако следует учитывать условия конкретной задачи и ограничения, накладываемые на объем памяти ОЗУ. Так как для агента очень важно, какие вычислительные мощности ему доступны, в частности какой объем вычислений может быть выполнен за один временно́й шаг, также сдерживающим фактором является доступная память [19].

Для преодоления проблемы размерности, связанной со значительным числом возможных состояний, предлагается воспользоваться компактной (относительно таблиц) параметризацией (приближенным методом решения) в виде аппроксимаций Q-функции полезности (рисунок 1) или π-функции стратегии (рисунок 2) искусственной нейронной сетью (ИНС) в виде многослойного персептрона (МСП) с параметрами w. Под глубокой нейронной сетью будем понимать ИНС, которая содержит три и более слоёв (два и более скрытых слоёв).

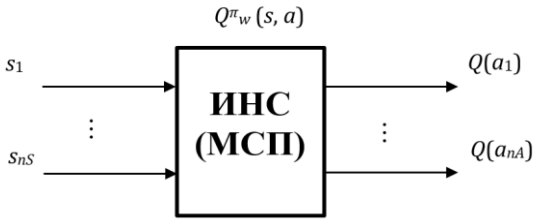


Рис. 1. Блок-схема аппроксиматора Q-функции полезности ИНС в виде МСП

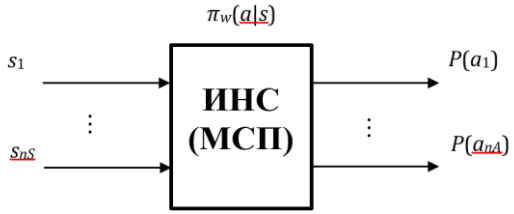


Рис. 2. Блок-схема аппроксиматора π-функции стратегии ИНС в виде МСП

При этом для глубокого обучения с подкреплением существует множество эффективных алгоритмов, основанных на полезности (SARSA, DQN), стратегии (Policy Gradients, REINFORSE), комбинированные методы (A2C, A3C) и др. [20]. Заметим, что для сокращения размерности выходного слоя ИНС (количества выходов МСП) целесообразно использовать подход, базирующийся на обучении с подкреплением при многоцелевых действиях агента [21].

Полученные значения действий агента (a_t) позволяют прогнозировать свободные (пригодные для радиосвязи) частотные каналы без разметки данных для обучения. Следует отметить, что если прогнозные значения находящихся рядом каналов «свободны» (равны «1»), то это позволяет выбрать соответствующую более широкую полосу частот на рабочей частоте (которая является центральной частотой в окрестности полосы).

3. Теоретическая оценка вычислительной сложности рассматриваемых алгоритмов. В работе рассматриваются алгоритмы следующие алгоритмы обучения с подкреплением: REINFORCE, SARSA, A2C. Перед проведением вычислительного эксперимента целесообразно оценить вычислительную эффективность каждого из них, а затем сопоставить теоретические оценки с полученным эмпирическим материалом. Формальное построение оценки

вычислительной эффективности произведено с учётом следующих ограничений:

1. Предполагается дискретное пространство действий (выбор одного из n каналов);
2. Агент в виде МСП содержит один скрытый слой;
3. Последнее выбранное действие (отклик) агента интегрировано в тензор состояния среды;
4. Вычисление функции активации будем считать не влияющий на асимптотическую сложность ($f(x) \overset{\text{Def}}{=} O(1)$).

Для построения оценки алгоритмов введём следующие обозначения:

1. Среда характеризуется размерностью состояния d_s; количеством дискретных действий d_a; макисмальной длинной эпизода T.
2. Агент (искусственная нейронная сеть) количеством нейронов в скрытом слое m; входной размерностью d_s; количеством выходных нейронов d_a.

Рассмотрим по порядку каждый из алгоритмов. Алгоритм REINFORCE представляет собой реализацию метода градиента политики (Policy Gradient), где агент обучает параметризованную политику $\pi_\theta(\mathrm{a}|\mathrm{s})$, реализованную как МСП путём максимизации ожидаемой награды. Политика генерирует действия и агрегируются траектории в рамках одного эпизода, вычисляются функция стоимости и градиент политики обновляется. Формула градиента:

$$\nabla_\theta J(\theta) = \mathbb{E}\tau \sim \pi\theta \left[\sum_{t=0}^{T} \nabla_\theta \, log\pi_\theta(a_t|s_t) \cdot G_t \right], \tag{11}$$

где $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$ дисконтированное вознаграждение (reward), γ – дисконтирующий фактор, $\tau = (s_0, a_0, r_0, \dots, s_T)$ – траектория в рамках эпизода. Для каждого эпизода вычисляется: прямое вычисление МСП для выбора действия $O(T \cdot m^2)$, вычисление функции вознаграждения $O(T)$ и вычисление градиентов для МСП $O(T \cdot m^2)$. Таким образом, основная вычисления приходятся на прямой проход и обратное распространение ошибки в МСП. Сложность алгоритма REINFORCE линейна по T, однако в значительной мере зависит от размера МСП. Для наших целей, можно считать, что вычисление REINFORCE заключается в вычислении прямого прохода МСП и обратного распространения ошибки по нему для каждой итерации эпизода.

Алгоритм SARSA представляет собой метод аппроксимации Q-функции $Q_\theta(s,a)$ реализованной в виде МСП. Агент выбирает действия по ε-жадной стратегии. Обновление на каждом шаге определяется следующим образом [20]:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\big[r_t + \gamma Q\big(s_{\{t+1\}},a_{\{t+1\}}\big) - Q(s_t,a_t)\big], \quad (12)$$

где $a_{t+1} \sim \pi(s_{t+1})$ представляет собой политику; α – скорость обучения. Градиенты для МСП в алгоритме SARSA вычисляются как [20]:

$$\theta = \theta - \alpha\nabla_\theta\big(r_t + \gamma Q_\theta(s_{t+1},a_{t+1}) - Q_\theta(s_t,a_t)\big)^2. \quad (13)$$

Аналогично алгоритму REINFORCE на первом этапе вычисляется прямой проход МСП $O(T\cdot m^2)$, после чего происходит шаг симуляции $O(T)$, затем второе вычисление прямого прохода по МСП для вычисления a_{t+1} $O(T\cdot \mathrm{m}^2)$, вычисление ошибки на основе функции вознаграждения $O(T)$ и вычисление градиентов для МСП $O(T\cdot \mathrm{m}^2)$. Таким образом, асимптотически алгоритм SARSA имеет какую же сложность что и REINFORCE, однако, как и в случае с REINFORCE можно дать оценку в количестве прямых проходов и обратного распространения ошибки, для SARSA соответственно 2 и 1 на каждую итерацию в эпизоде.

Последний рассматриваемый алгоритм A2C представляет собой синхронный вариант метода актор-критик и имеет 2 МСП: актора для аппроксимации политики $\pi_\theta(\mathrm{a}|\mathrm{s})$ и критика для оценки функции стоимости $V_\varphi(s)$. Также как и в предыдущих алгоритмах, на первом этапе вычисляется прямой проход по МСП-актору $O(T\cdot \mathrm{m}^2)$, затем вычисляется прямой проход по МСП-критику $O(T\cdot \mathrm{m}^2)$, после чего вычисляются шаг симуляции и функция потерь $O(T)$, завершается все вычислением градиентов для МСП-актора $O(T\cdot \mathrm{m}^2)$ и МСП-критика $O(T\cdot \mathrm{m}^2)$. Таким образом, для алгоритма A2C сложность линейна по длине эпизода, однако на каждую итерацию необходимо вычислить 2 прямых прохода по МСП и 2 прохода обратного распространения ошибки.

Учитывая тот факт, что наибольший вклад в оценку вычислительной сложности рассматриваемых алгоритмов вносит прямой проход и обратное распространение ошибки в МСП целесообразно расширить оценку на произвольный МСП с L скрытыми слоями по $m \in \{m_1, m_2, \dots, m_L\}$ нейронов в каждом, таким образом сняв

ограничение II. Учитывая ограничение IV прямой проход по l слою L- слойного МСП оценивается как:

$$O\left(m_{\{l-1\}} \times m_{\{l\}} + m_{\{l\}}\right) \to O\left(m_{\{l-1\}} \times m_{\{l\}}\right) \xrightarrow[m=max(m_{l-1},m_l)]{} O(m^2). \quad (14)$$

Заметим, что сложность прямого прохода в терминах O равна сложности обратного распространения ошибки. В качестве верхней оценки вычислительной сложности для L-слойного МСП целесообразно принять значение $O(L \times \max(m)^2)$ – сложность вычисления каждого слоя соответствует сложности вычисления наиболее широкого (содержащего наибольшее число нейронов) слоя МСП. В рамках рассматриваемой задачи данное огрубление оценки допустимо, так как, сравниваются алгоритмы, обучающие одинаковые L- слойные МСП (в том числе МСП-критик для алгоритма SARSA имеет ту же архитектуру, что и МСП для остальных алгоритмов). Таким образом, для всех рассмотренных алгоритмов сложность остаётся линейной по T, оценка не изменяется при замене реализации аппроксиматора с однослойного персептрона на МСП.

В результате проведённого анализа, учитывая, что проход обратного распространения ошибки по МСП содержит, в среднем в 2 раза больше операций [4], сформулируем следующие количественные оценки: наиболее быстрым является алгоритм REINFORCE, приблизительно в 1,5 раза медленнее SARSA, а A2C приблизительно в 2 раза медленнее.

4. Моделирование среды. Для моделирования работы алгоритмов обучения с подкреплением на основе формализации задачи была построена следующая модель среды: в наборе из N каналов в каждый момент времени t свободен только один канал. Формально симуляция среды представляет собой правило, по которому для каждого шага времени t строится вектор из N величин соответствующих уровню шумов, индекс элемента вектора соответствует индексу канала. Среда является параметрической и определяется следующим набором параметров: N – общее число каналов, $f_{\text{пор}}(t)$ – закон изменения свободного канала; $P_{(\text{ш},min)}$ – минимальный уровень шумов (дБ); $P_{(\text{ш},max)}$ – максимальный уровень шумов (дБ); $P_{(c,min)}$ – минимальный уровень полезного сигнала (дБ); $P_{(c,max)}$ – максимальный уровень полезного сигнала (дБ); V – максимальная амплитуда флуктуаций шумов; T – общее время (дискретное) генерации, сколько векторов будет сгенерировано до окончания текущей сессии.

Подобная формализация позволяет достаточно гибко моделировать сложные ситуации, например при $P_{(c,min)} < P_{(\text{ш},max)}$ в некоторых моментах времени t полезный сигнал может оказаться ниже уровня шумов и т.д. На рисунке 3 приведена визуализация для среды с периодическим законом $f_{\text{пор}}$ для числа каналов $N = 3$ и $N = 15$.

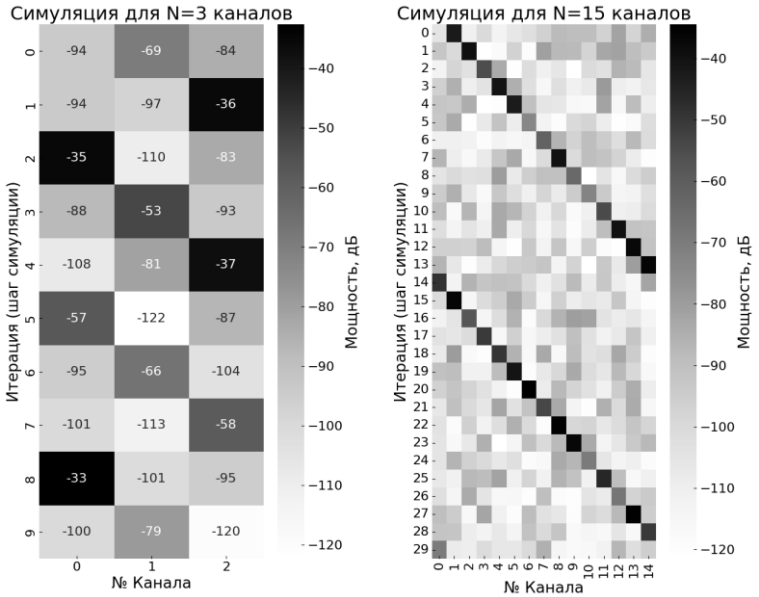


Рис. 3. Визуализация среды для $N = 3$ и $N = 15$ каналов

Порождающий закон является детерминированным и периодическим (для обеспечения генерации любого наперёд заданного количества шагов генерации T), с этой точки зрения среда является детерминированной. С другой стороны, выбор конкретных числовых значений на каждом шаге происходит по случайному закону. Рассмотрим формирование вектора состояния среды X^t на шаге t подробно. При инициализации среды для каждого канала выбирается значение из равномерного распределения $X^0 = \{x_i = U[P_{(\text{ш},min)}, P_{(\text{ш},max)}] \forall i \in N\}$ – устанавливается «фон». Далее, определяется индекс свободного канала на шаге $t = 0$ из порождающего закона $n = f_{\text{пор}}(0)$, значение $X_n^0 = U[P_{(c,min)}, P_{(c,max)}]$. Далее рассмотрим переход $t_1 \to t$: для фонового шума вычисляется изменение $X^t = \left\{x_i^{(t-1)} = x_i^{(t-1)} + U[-V, V] \forall i \in N\right\}$,

если рассчитанная величина x_i^t превышает $P_{(\text{ш},max)}$ или ниже $P_{(\text{ш},min)}$ то знак выражения меняется $\left(x_i^{(t-1)} = x_i^{(t-1)} - U[-V,V]\right)$. Таким образом, обеспечивается плавное изменение фоновых шумов. Для свободного канала операция $t_1 \to t$ аналогичная инициализации.

Технически для того, чтобы обеспечить плавное изменение уровня шума в программном классе генерации хранится вектор состояния фона (без свободного канала) и полный вектор среды раздельно. Однако такие накладные расходы в современных компьютерах пренебрежительно малы.

Предложенная среда позволяет генерировать достаточно сложные последовательности состояний, например, $f_{\text{пор}}$ может быть непериодическим или случайным. В настоящей работе был рассмотрен случай периодического порождающего закона, обобщение результатов на более широкий класс порождающих законов является предметом дальнейших работ.

5. Обсуждение результатов. В качестве примера рассмотрим среду, в которой два из $N = 3$ радиоканалов заняты, а один свободен. Положим, что номер свободного канала изменяется по детерминированному закону с периодически повторяющейся последовательностью на 30 временных интервалах. Состояние радиоканала моделируется следующими уровнями: минимальный уровень шумов -120 дБ, максимальный уровень шумов -80 дБ, минимальный уровень полезного сигнала -80 дБ, максимальный уровень полезного сигнала -120 дБ. Амплитуда флуктуации шумов 3 дБ, порождающий закон $f_{(\text{пор})}(t) = mod(t, N)$ – переодическое смещение «вправо» по каналам (по модулю N).

В качестве аппроксимации политики используется следующий набор МСП: с одним скрытым слоем, содержащим 10 нейронов; с двумя скрытыми слоями по 10 нейронов в каждом; с тремя скрытыми слоями по 10 нейронов в каждом.

Для каждого МСП из набора проводилась серия экспериментов по обучению алгоритмами REINFORCE, SARSA и A2C.

В рассмотренном для численного моделирования примере награда агента выбиралась двумя способами – в первом r_{t+1} зависит от состояния в следующий наблюдаемый момент времени, следующим образом:

$$r_{t+1} = \begin{cases} 1, & \text{PdB}_k \le -80; \\ -\left(\dfrac{\text{PdB}_k}{50} + 1{,}6\right), & \text{PdB}_k > -80, \end{cases} \tag{15}$$

где PdB_k – мощность (дБ) в выбранном (предсказанном) канале связи. Во втором – на основе выражения (4).

Для первого способа вознаграждения на основе выражения (14) на рисунке 4 представлен график для метода градиента стратегии (политики) по алгоритму REINFORSE зависимости суммарных (полных) вознаграждений и скользящее среднее по оценкам в 200 контрольных точках. Следует отметить, что максимальное полное вознаграждение составляет 30.

Оптимизация производилась с помощью метода Adam [22] со скорость обучения 0,001; параметры методов обучения с подкреплением: начальное значение $\varepsilon = 0{,}5$; конечное значение $\varepsilon = 0{,}001$; коэффициент дисконтирования $\gamma = 0{,}7$. На рисунках 4-9 используется сокращение «СВ» – суммарное вознаграждение, цифра после МСП («МСП-1») обозначает количество скрытых слоёв в МСП.

В ходе экспериментов алгоритм REINFORCE демонстрирует способность достигать оптимального (безошибочного) решения в диапазоне от 10000-12000 эпизодов обучения. Однако стоит отметить, что траектория функции вознаграждения характеризуется заметными осцилляциями, особенно на поздних этапах обучения.

На рисунке 5 представлен график для метода градиента полезности (ценности) по алгоритму SARSA зависимости суммарных (полных) вознаграждений и соответствующее скользящее среднее для способа вознаграждения на основе выражения (15).

Алгоритм SARSA, демонстрирует достижение оптимального (безошибочного) решения в районе 8000 эпизодов обучения. Такая относительно высокая скорость сходимости по сравнению REINFORCE, может быть объяснена фундаментальными особенностями SARSA: механизмом самонастройки (bootstrapping), где обновление Q-значений для текущей пары (состояние, действие) зависит от оценки следующего действия, выбранного в соответствии с той же политикой. В частности, правило обновления SARSA подразумевает, что агент оценивает текущее действие на основе предсказанной ценности следующего состояния и действия $Q(s', a')$, где a' выбирается не оптимально (как в Q-Learning), а по текущей, возможно, случайной, политике – например, на ранних этапах ε-жадной стратегии выбора действий. С другой стороны, после снижения ε до практически нулевых значений (около 8000 эпизодов) дополнительная оценка ценности следующего действия приводит к более быстрой сходимости к безошибочному решению и снижению осцилляций графика суммарного вознаграждения (которые обусловлены ненулевым значением ε на протяжении всего обучения).

Аналогичной особенностью обусловлена и повышенная дисперсия функции суммарного вознаграждения, что представлено на рисунке 4: поскольку обновления опираются на коррелированные образцы из траекторий, генерируемых текущей политикой, случайные колебания в выборе действий (включая случайные шаги на начальных итераций ε-жадной стратегии выбора действий) усиливают шум в оценках, приводя к более высокой амплитуде колебаний функции.

На рисунке 6 представлен график для комбинированного метода (актор-критика с преимуществом) по алгоритму A2C зависимости суммарных (полных) вознаграждений и соответствующее скользящее среднее для способа вознаграждения на основе выражения (14).

Алгоритм A2C), демонстрирует достижение оптимального (безошибочного) решения около 8000 эпизодов обучения. Скорость сходимости аналогична SARSA, однако порождена другим механизмом. Она, в первую очередь, обусловлена использованием двух нейронных сетей: МСП-актора для аппроксимации политики $\pi(a|s)$ и МСП-критика для оценки функции ценности $V(s)$, которая прогнозирует ожидаемое дисконтированное суммарное вознаграждение по текущему состоянию. На основе этой оценки производится более эффективная оптимизация по сравнению с алгоритмами, строящих оценки на основе метода Монте-Карло. Однако, как было показано в разделе 4, алгоритм A2C в 2 раза медленнее REINFOCE, следовательно, быстрая сходимость по эпизодам компенсируется высокой вычислительной сложностью каждой эпизода. Этим же объясняется наличии больших (по амплитуде и частоте) осцилляций графика суммарного вознаграждения по сравнению с SARSA, так как МСП-критик даже на поздних эпохах обучения может производить неверные прогнозы функции ценности $V(s)$.

Анализ графиков, представленных на рисунках 3-5, позволяет сделать вывод о том, что алгоритмы SARSA и A2C достигают полного (безошибочного) предсказания свободного канала в модельной среде за одинаковое (в среднем) количество эпизодов. Алгоритм REINFORCE не достигает уровня полного (безошибочного) предсказания. Следует отметить, что графики скользящего среднего получены по серии экспериментов (по 10 для каждого алгоритма и каждой реализации МСП) и отражают общий характер, а не конкретную реализацию, траектория обучения которой достаточно сильно зависит от начальной инициализации весов МСП.

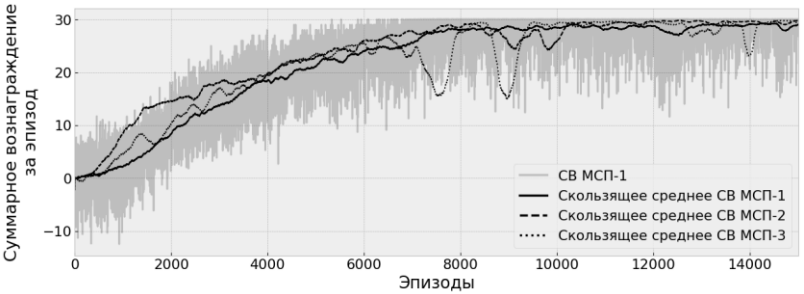


Рис. 4. Полное вознаграждение и скользящее среднее для алгоритма REINFORSE

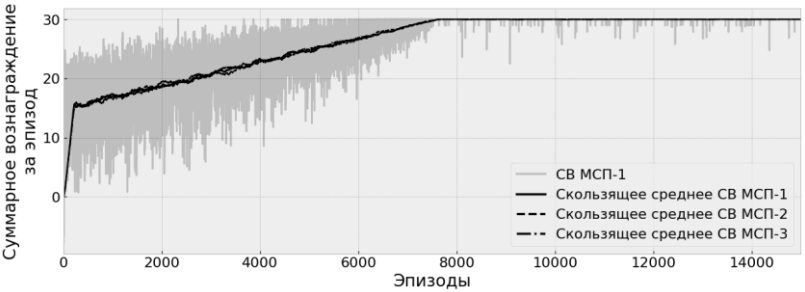


Рис. 5. Полное вознаграждение и скользящее среднее для алгоритма SARSA

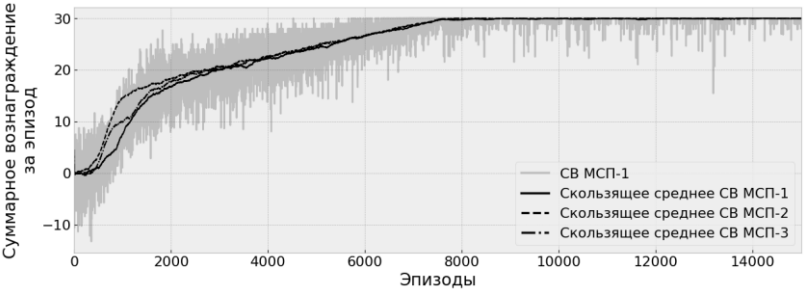


Рис. 6. Полное вознаграждение и скользящее среднее для алгоритма A2C

Для второго способа вознаграждения на основе выражения (4) на рисунках 7-9 представлены графики соответственно для алгоритмов REINFORSE, SARSA и A2C зависимости суммарных (полных) вознаграждений и скользящее среднее.

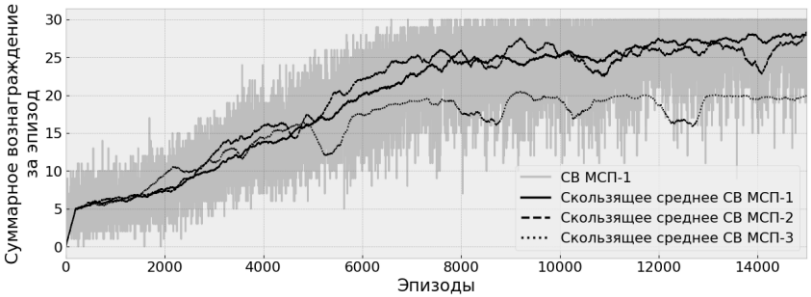


Рис. 7. Полное вознаграждение и скользящее среднее для алгоритма REINFORSE

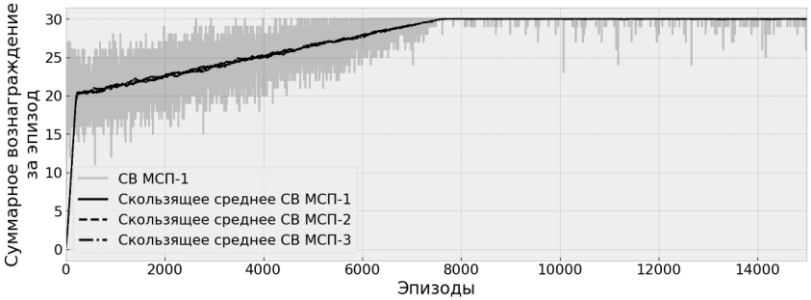


Рис. 8. Полное вознаграждение и скользящее среднее для алгоритма SARSA

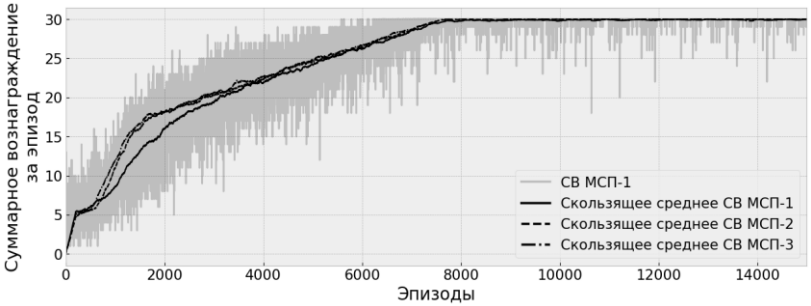


Рис. 9. Полное вознаграждение и скользящее среднее для алгоритма A2C

Из сравнения графиков на рисунках 7-9 и рисунках 4-6 следует, что выбор конкретной формализации функции вознаграждения (4) или (15) не оказывает существенного влияния на общую сходимость рассмотренных алгоритмов SARSA и A2C, однако функция (15) значительно улучшает сходимость алгоритма REINFORCE.

На рисунке 10 приведены графики среднего времени вычисления одного эпизода в разрезе алгоритмов и реализаций МСП. Следует отметить, что все эксперименты проводились в одинаковых условиях (зафиксирована программная реализация библиотеки вычисления градиентов, аппаратная составляющая и программная реализация моделируемой среды).

На рисунке 10 высота столбца соответствует среднему времени вычисления одного эпизода обучения; черные линии – стандартное отклонение 3σ (99,7% эпизодов). Из рисунка 9 следует, что оценки приведённые в разделе 4 подтверждаются вычислительным экспериментом: алгоритм REINFORCE является наиболее быстрым, SARSA медленнее (в среднем) в 1,67 раз, A2C медленнее относительно REINFORCE (в среднем) в 2,10 раз.

Другим важным выводом является тот факт, что формализации функции вознаграждения (4) или (15) не оказывает значительного влияния на сходимость алгоритмов SARSA и A2C, однако формализация (15) является более универсальной и повышает эффективность алгоритма REINFORCE. Таким образом, формализация (15) может быть рекомендована как отправная точка в решении практических задач выбора свободного канала.

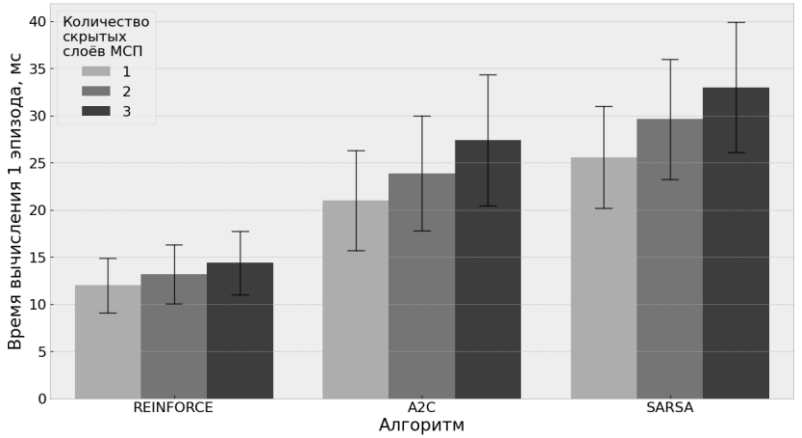


Рис. 10. Зависимость среднего суммарного вознаграждения от времени

На рисунке 11 приведён сводный график для обучения агентов алгоритмом REINFORECE для $N \in [3{,}100]$. Все внутренние параметры среды и агентов фиксированы (численные значения совпадают со значениями, приведёнными в начале раздела 5 для $N = 3$). Для

каждого N выбирается число временных интервалов как $T = N \times 10$, таким образом, для детально рассмотренного случая $N = 3: T = 30$, для $N = 100: T = 1000$ и т.д.

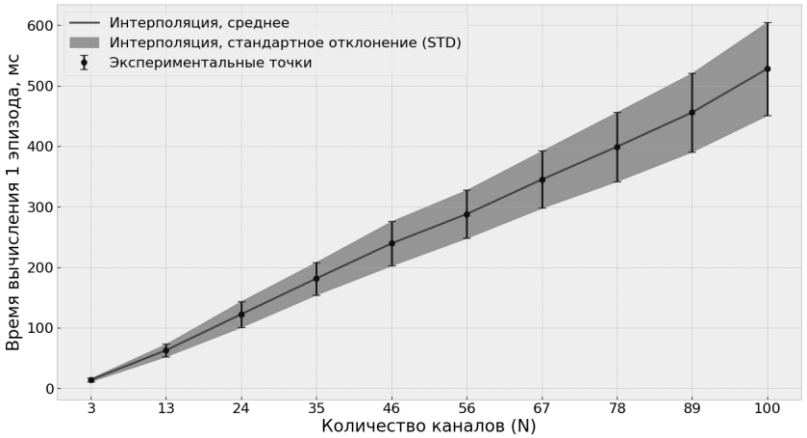


Рис. 11. Сводный график обучения агентов для $N \in [3{,}100]$

Следует отметить, что время растёт приблизительно как $O(N)$, однако, подобная тенденция должна быть исследована в областях больших ($N > 10^3$), так как вероятно, увеличение сложности изменит своё поведение. Подобное исследование целесообразно при практической заинтересованности в агентах, работающих с тысячами каналов. С другой стороны, для рассматриваемой модели среды обеспечивается устойчивое обучение агента до безошибочного решения в рассмотренном диапазоне каналов. Таким образом, можно сделать вывод о достаточно эффективном обобщении решения задачи на большее число каналов, при учёте сохранения параметров модели среды.

6. Заключение. Таким образом, для решения задачи формирования прогноза рабочих частот и полос частотной области предложен метод для системы когнитивного радио, базирующийся на глубоком обучении с подкреплением. Предложенный конструктивный метод позволяет оценить рабочие частоты и соответствующую ширину полосы частот при заданных условиях и ограничениях. Учитывая теоретические оценки вычислительной сложности, полученные в разделе 4 и анализируя отношения времени сходимости алгоритмов до безошибочного решения можно сделать вывод о согласованности теоретических и экспериментальных

результатов. Целью дальнейших исследований целесообразно определить анализ оценки эффективности включения дополнительной априорной информации в матрицу состояний среды и выбор функций вознаграждения для решения конкретных научно-прикладных задач когнитивного радио.

Литература

1. Тонг В., Чжу П. Сети 6G. Путь от 5G к 6G глазами разработчиков. От подключенных людей и вещей к подключенному интеллекту // М.: ДМК Пресс. 2022. 624 с.
2. Fette B.-A. Cognitive Radio Technology // Elsevier. 2006. 622 p.
3. Комашинский В.И., Смирнов Д.А. Нейронные сети и их применение в системах управления и связи // М.: Горячая линия–Телеком. 2003. 93 с.
4. Голубинский А.Н., Толстых А.А. Гибридный метод обучения сверточных нейронных сетей // Информатика и автоматизация. 2021. Т. 20. № 2. С. 463–490. DOI: 10.15622/ia.2021.20.2.8.
5. Wu C., Chowdhury K.-R., Di Felice M., Meleis M. Spectrum Management of Cognitive Radio Using Multi-Agent Reinforcement Learning // 9th International Conference on Autonomous Agents and Multiagent Systems. 2010. vol. 1–3. pp. 1705–1712. DOI: 10.1145/1838194.1838199.
6. Kiran U., Kumar P.-D., Reddy R.-K., Ranjith M. Efficient Exploration for Reinforcement Learning Based Distributed Spectrum Sharing in Cognitive Radio System // International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2013. vol. 2. no. 11. pp. 5596–5604.
7. Yau K.-L.-A., Poh G.-S., Chien S.-F., Al-Rawi1 H.-A.-A. Application of Reinforcement Learning in Cognitive Radio Networks: Models and Algorithms // The Scientific World Journal. 2014. vol. 1. pp. 1–23. DOI: 10.1155/2014/209810.
8. Abolarinwa J.-A., Latiff A.-N.-M. Channel Decision in Cognitive Radio Enabled Sensor Networks A Reinforcement Learning Approach // International Journal of Engineering and Technology (IJET). 2015. vol. 7. no. 4. pp. 1394–1404.
9. Raj V., Dias I., Tholeti T., Kalyani S. Spectrum Access In Cognitive Radio Using A Two Stage Reinforcement Learning Approach // IEEE. 2018. vol. 12. no. 1. pp. 20–34. DOI: 10.1109/JSTSP.2018.2798920.
10. Tubachi S., Venkatesan M., Kulkarni A.-V., et al. Predictive learning model for Cognitive Radio using Reinforcement Learning // IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). 2017. pp. 564–567. DOI: 10.1109/ICPCSI.2017.8391775.
11. Jang S.-J., Han C.-H., Lee K.-E., et al. Reinforcement learning-based dynamic band and channel selection in cognitive radio ad-hoc networks // J Wireless Com Network. 2019. vol. 2019. pp. 1–25. DOI: 10.1186/s13638-019-1433-1.
12. Singhal C., Thanikaiselvan V. Cross Layering Using Reinforcement Learning In Cognitive Radio-Based Industrial Internet Of Ad-Hoc // International Journal of Computer Networks & Communications (IJCNC). 2022. vol. 14. no. 4. pp. 1–17. DOI: 10.5121/ijcnc.2022.14401.
13. Talekar S., Banait S., Patil M. Improved Q-Reinforcement Learning Based Optimal Channel Selection In Cognitive Radio Networks // International Journal of Computer Networks & Communications (IJCNC). 2023. vol. 15. no. 3. pp. 1–14. DOI: 10.5121/ijcnc.2023.15301.
14. Rosen D., Rochez I., McIrvin C., Lee J., D'Alessandro K., Wiecek M., et al. RFRL Gym_A Reinforcement Learning Testbed for Cognitive Radio Applications //

International Conference on Machine Learning and Applications (ICMLA). 2023. pp. 279–286. DOI: 10.1109/ICMLA58977.2023.00046.
15. Отчет (Сектора связи Международного союза электросвязи) МСЭ-R SM.2152 (09/2009) «Определения системы радиосвязи с программируемыми параметрами (SDR) и системы когнитивного радио (CRS)».
16. Chitnavis S., Kwasinski A. Cross Layer Routing in Cognitive Radio Network Using Deep Reinforcement Learning // IEEE Wireless Communications and Networking Conference (WCNC). 2019. pp. 1–13. DOI: 10.1109/WCNC.2019.8885918.
17. Obite F., Usman A.-D., Okafor E. An overview of deep reinforcement learning for spectrum sensing in cognitive radio networks // Digital Signal Processing. 2021. vol. 113. pp. 1–18. DOI: 10.1016/j.dsp.2021.103014.
18. Tondwalkar A., Kwasinski A. Deep Reinforcement Learning for Distributed and Uncoordinated Cognitive Radios Resource Allocation. 2022. pp. 1–13. arXiv: 2205.13944v1.
19. Саттон Р. С., Барто Э. Дж. Обучение с подкреплением // М.: ДМК Пресс. 2020. 552 с.
20. Грессер Л., Кенг В.Л. Глубокое обучение с подкреплением: теория и практика на языке Python // СПб.: Питер. 2022. 416 с.
21. Wang H., Yu Y. Exploring Multi-Action Relationship in Reinforcement Learning // Springer, Cham. 2016. pp. 1–13. DOI: 10.1007/978-3-319-42911-3_48.
22. Kingma D., Ba J. Adam: A Method for Stochastic Optimization. // CoRR. 2014. T. abs/1412.6980.

Толстых Андрей Андреевич — канд. техн. наук, инженер-программист, ООО «РТК». Область научных интересов: искусственные нейронные сети, машинное обучение, обучение с подкреплением. Число научных публикаций — 65. tolstykh.aa@yandex.ru; проспект Высоковольтный, 1, 127566, Москва, Россия.

Голубинский Андрей Николаевич — д-р техн. наук, доцент, начальник отдела, Российский научный фонд (РНФ). Область научных интересов: машинное обучение, нейросетевое моделирование, автоматизированные системы управления с элементами искусственного интеллекта, обработка речевых сигналов. Число научных публикаций — 250. annikgol@mail.ru; улица Солянка, 14, 109240, Москва, Россия; р.т.: +7 (910) 346-6537.

DOI 10.15622/ia.25.2.2

A. TOLSTYKH, A. GOLUBINSKIY

DATA MINING BASED ON DEEP REINFORCEMENT LEARNING FOR PREDICTION OF OPERATING FREQUENCIES AND BANDS IN A COGNITIVE RADIO SYSTEM

Tolstykh A., Golubinskiy A. **Data Mining Based on Deep Reinforcement Learning for Prediction of Operating Frequencies and Bands in a Cognitive Radio System.**

Abstract. The paper proposes a method for solving the problem of choosing a communication channel in cognitive radio based on information about the current state of all available communication channels using the mathematical apparatus of reinforcement learning. The method consists in formalizing the problem of choosing communication channels in terms of "environment-agent" and training agents using the REINFORCE, SARSA and A2C algorithms. The calculation of memory costs for solving the problem of selecting communication channels using classical methods is given. The memory estimate is $\mathbf{4\times2^{2n}}$ bytes for a random state of channels (busy/free) and $\mathbf{4\times n^2}$ bytes for one free channel at each step when solving the problem using the tabular Q-learning algorithm. Two different formalizations of the reward for the agent within the framework of the problem being solved using reinforcement learning are presented – for the trivial case (binary availability/unavailability of the frequency channel) and for a more complex case considering the power (in dB) in the selected communication channel. The restriction on the first formalization is that at each iteration there should be only one free communication channel out of all available channels. The second proposed formalization of the reward function does not impose such restrictions and is more universal. Computational experiments are presented for the corresponding formalizations of the reward function. Agents are trained using the SARSA and A2C algorithms. On average, error-free solutions are achieved after 8,000 training episodes for the corresponding formalizations of training in a model problem for various agent implementations. The REINFORCE algorithm does not provide error-free solutions, but reward formulation takes into account the improved training efficiency of the REINFORCE algorithm. Theoretical estimates of the computational complexity of the considered methods are provided, which are consistent with the computational experiments.

Keywords: cognitive radio, reinforcement learning, deep learning, artificial neural network, multilayer perceptron, reward function, software-defined radio, synthetic data, augmentation, artificial intelligence.

References

1. Tong W., Zhu P. Seti 6G. Put' ot 5G k 6G glazami razrabotchikov. Ot podklyuchennykh lyudey i veshchey k podklyuchennomu intellektu [6G networks. The path from 5G to 6G through the eyes of developers. From connected people and things to connected intelligence]. Moscow: DMK Press, 2022. 624 p. (In Russ.).
2. Fette B.-A. Cognitive Radio Technology. Elsevier. 2006. 622 p.
3. Komashinskiy V., Smirnov D. Neyronnye seti i ikh primenenie v sistemakh upravleniya i svyazi [Neural networks and their application in control and communication systems]. Moscow: Hotline–Telecom, 2003. 93 p. (In Russ.).
4. Golubinskiy A., Tolstykh A. [Hybrid method for training convolutional neural networks]. Informatika i avtomatizatsiya – Informatics and Automation. 2021. vol. 20. no. 2. pp. 463–490. DOI: 10.15622/ia.2021.20.2.8. (In Russ.).
5. Wu C., Chowdhury K.-R., Di Felice M., Meleis M. Spectrum Management of Cognitive Radio Using Multi-Agent Reinforcement Learning. 9th International Conference

on Autonomous Agents and Multiagent Systems. 2010. vol. 1–3. pp. 1705–1712. DOI: 10.1145/1838194.1838199.
6. Kiran U., Kumar P.-D., Reddy R.-K., Ranjith M. Efficient Exploration for Reinforcement Learning Based Distributed Spectrum Sharing in Cognitive Radio System. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2013. vol. 2. no. 11. pp. 5596–5604.
7. Yau K.-L.-A., Poh G.-S., Chien S.-F., Al-Rawi1 H.-A.-A. Application of Reinforcement Learning in Cognitive Radio Networks: Models and Algorithms. The Scientific World Journal. 2014. vol. 1. pp. 1–23. DOI: 10.1155/2014/209810.
8. Abolarinwa J.-A., Latiff A.-N.-M. Channel Decision in Cognitive Radio Enabled Sensor Networks A Reinforcement Learning Approach. International Journal of Engineering and Technology (IJET). 2015. vol. 7. no. 4. pp. 1394–1404.
9. Raj V., Dias I., Tholeti T., Kalyani S. Spectrum Access In Cognitive Radio Using A Two Stage Reinforcement Learning Approach. IEEE. 2018. vol. 12. no. 1. pp. 20–34. DOI: 10.1109/JSTSP.2018.2798920.
10. Tubachi S., Venkatesan M., Kulkarni A.-V., et al. Predictive learning model for Cognitive Radio using Reinforcement Learning. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). 2017. pp. 564–567. DOI: 10.1109/ICPCSI.2017.8391775.
11. Jang S.-J., Han C.-H., Lee K.-E., et al. Reinforcement learning-based dynamic band and channel selection in cognitive radio ad-hoc networks. J Wireless Com Network. 2019. vol. 2019. pp. 1–25. DOI: 10.1186/s13638-019-1433-1.
12. Singhal C., Thanikaiselvan V. Cross Layering Using Reinforcement Learning In Cognitive Radio-Based Industrial Internet Of Ad-Hoc. International Journal of Computer Networks & Communications (IJCNC). 2022. vol. 14. no. 4. pp. 1–17. DOI: 10.5121/ijcnc.2022.14401.
13. Talekar S., Banait S., Patil M. Improved Q-Reinforcement Learning Based Optimal Channel Selection In Cognitive Radio Networks. International Journal of Computer Networks & Communications (IJCNC). 2023. vol. 15. no. 3. pp. 1–14. DOI: 10.5121/ijcnc.2023.15301.
14. Rosen D., Rochez I., McIrvin C., Lee J., D'Alessandro K., Wiecek M., et al. RFRL Gym_A Reinforcement Learning Testbed for Cognitive Radio Applications. International Conference on Machine Learning and Applications (ICMLA). 2023. pp. 279–286. DOI: 10.1109/ICMLA58977.2023.00046.
15. Report (International Telecommunication Union Telecommunication Sectors) ITU-R SM.2152 (09/2009) [Definitions of Software Defined Radio (SDR) and Cognitive Radio System (CRS)]. (In Russ.).
16. Chitnavis S., Kwasinski A. Cross Layer Routing in Cognitive Radio Network Using Deep Reinforcement Learning. IEEE Wireless Communications and Networking Conference (WCNC). 2019. pp. 1–13. DOI: 10.1109/WCNC.2019.8885918.
17. Obite F., Usman A.-D., Okafor E. An overview of deep reinforcement learning for spectrum sensing in cognitive radio networks. Digital Signal Processing. 2021. vol. 113. pp. 1–18. DOI: 10.1016/j.dsp.2021.103014.
18. Tondwalkar A., Kwasinski A. Deep Reinforcement Learning for Distributed and Uncoordinated Cognitive Radios Resource Allocation. 2022. pp. 1–13. arXiv: 2205.13944v1.
19. Sutton R., Barto E. Obuchenie s podkrepleniem [Reinforcement learning]. Moscow: DMK Press, 2020. 552 p. (In Russ.).
20. Gresser L., Keng V. Glubokoye obuchenie s podkrepleniem: teoria i praktika na yazyke Python [Deep Reinforcement Learning: Theory and Practice in Python]. SPB: Peter. 2022. 416 p.

21. Wang H., Yu Y. Exploring Multi-Action Relationship in Reinforcement Learning. Springer, Cham. 2016. pp. 1–13. DOI: 10.1007/978-3-319-42911-3_48.
22. Kingma D., Ba J. Adam: A Method for Stochastic Optimization. CoRR. 2014. T. abs/1412.6980.

Tolstykh Andrey — Ph.D., Software engineer, OOO “RTK”. Research interests: artificial neural networks, machine learning, reinforcement learning. The number of publications — 65. tolstykh.aa@yandex.ru; 1, Vysokovoltny Ave., 127566, Moscow, Russia.

Golubinskiy Andrey — Ph.D., Dr.Sci., Associate Professor, Head of Department, Russian Science Foundation (RSF). Research interests: machine learning, neural network modeling, automated control systems with artificial intelligence elements, speech signal processing. The number of publications — 250. annikgol@mail.ru; 14, Solyanka St., 109240, Moscow, Russia; office phone: +7 (910) 346-6537.

УДК 004.9 **DOI** 10.15622/ia.25.2.3

А.С. Миронов, А.А. Саенко, Е.С. Фомина

РАЗРАБОТКА АРХИТЕКТУРЫ ЦИФРОВОГО ДВОЙНИКА ЭКОСИСТЕМЫ ВОДНОГО ОБЪЕКТА ДЛЯ СОХРАНЕНИЯ ЭКОЛОГИЧЕСКОЙ УСТОЙЧИВОСТИ

Миронов А.С., Саенко А.А., Фомина Е.С. **Разработка архитектуры цифрового двойника экосистемы водного объекта для сохранения экологической устойчивости.**

Аннотация. Целью работы является разработка архитектуры цифрового двойника экосистемы водного объекта для обеспечения экологической устойчивости, включая мониторинг, прогнозирование и управление водными ресурсами в условиях антропогенного воздействия и климатических изменений. В работе предложена модульная архитектура цифрового двойника, включающая модуль сбора и слияния мультимодальных данных, цифровые модели экосистемы, экспертную систему, модуль регулирования нормативно-правовой базы, геоинформационную систему, модуль графического представления данных. Архитектура основана на интеграции данных с сенсоров, спутников, метеостанций и систем дистанционного зондирования, с применением моделей машинного обучения и больших языковых моделей для анализа нормативной документации. Разработана архитектура цифрового двойника водного объекта, позволяющая в реальном времени отслеживать параметры экосистемы, оценивать соответствие нормативам, формировать отчёты и рекомендации для субъектов природопользования и органов управления. Проведено тестирование эффективности различных моделей обработки правовых текстов, выявлены оптимальные подходы к их семантическому анализу. Обоснована возможность интеграции цифрового двойника в систему государственного регулирования природных ресурсов. Предложенная авторами архитектура цифрового двойника представляет собой комплексный инструмент устойчивого управления водными ресурсами, обеспечивающий прогнозирование состояния водных объектов, своевременное выявление рисков и формирование научно обоснованных управленческих решений, способствуя снижению экологических угроз и сохранению водных ресурсов.

Ключевые слова: цифровой двойник, водные ресурсы, экосистема водного объекта, управление природными ресурсами, экспертная система.

1. Введение. Природные экосистемы и окружающая среда во всем мире испытывают существенное давление из-за комплекса глобальных и локальных факторов: изменения климата, рост населения, урбанизация, загрязнение и нерациональное использование природных ресурсов [1]. Водные объекты играют фундаментальную роль в поддержании климатического баланса, гидрологического цикла и здоровья человека, однако они подвержены быстрой деградации под воздействием антропогенных факторов.

Примеры из разных регионов мира иллюстрируют масштаб и разнообразие вызовов, с которыми сталкиваются водные объекты. Так, в Китае в северо-западном оазисе Миньцинь [2] отмечается критическое снижение уровня грунтовых вод и деградация экосистем, что негативно сказывается на социально-экологических системах региона.

Реализованные меры по комплексному управлению бассейном реки Шиян привели к стабилизации уровня грунтовых вод и частичному восстановлению экологии, однако сохранение экосистем требует продолжительной адаптивной стратегии с учётом внутренних региональных особенностей. Аналогичные проблемы наблюдаются в озере Илун [3], которое за последние десятилетия испытало значительное сокращение площади, вызванное мелиоративными работами и изменениями в землепользовании. Бассейн реки Хуанхэ сталкивается с серьезными экологическими проблемами, включая дефицит воды, эрозию почв, снижение биоразнообразия и деградацию водных и лесных экосистем, что усугубляется антропогенными и климатическими воздействиями. Подобные вызовы так же фиксируются и в бассейне реки Хунцзян, где наблюдается ухудшение качества воды, утрата биоразнообразия и усиление эрозии почв, что усугубляется промышленным и бытовым загрязнением, а также деградацией лесных сообществ. Комплексные природоохранные проекты, включающие улучшение очистки сточных вод и увеличение лесных площадей, демонстрируют положительный эффект, приводящий к восстановлению экосистем.

Особенно важна значимость гидрологического режима как ключевого фактора устойчивости экосистем водных объектов. В бассейне реки Амур [4], на территории России и Китая, наблюдается сокращение водных поверхностей и водной растительности вследствие климатических изменений и антропогенного воздействия. В озере Цаохай провинции Гуйчжоу резкие колебания уровня воды, а не эвтрофикация, вызвали массовое вымирание погружённой растительности, что подчёркивает необходимость регулирования гидрологического режима [5]. Озеро Тайху в провинции Цзянсу стало объектом масштабной вспышки цветения воды, что привело к серьезным экологическим угрозам и заставило власти принять системные меры по улучшению качества воды и восстановлению природных экосистем [6]. Один из наиболее показательных примеров – озеро Байяндянь, крупнейшее мелководное озеро севера Китая [7]. После создания в 2017 году нового района Сюнъань были инициированы масштабные мероприятия по восстановлению водной среды озера. Только в 2023 году удалось наблюдалось резкое увеличение биомассы погружённых макрофитов в 4,2 раза по сравнению с 2018 годом, что свидетельствует о переходе от доминирования водорослей к макрофитному режиму. Построение пищевой сети с использованием модели Ecopath с Ecosim подтвердило перестройку трофических связей и увеличение устойчивости системы.

Основными факторами восстановления стали снижение нагрузки питательными веществами, гидрологическая регуляция и биологическое управление.

В европейских странах также зафиксированы множественные случаи деградации и восстановления водных экосистем. В бассейне реки Фийос отмечается высокая вероятность катастрофических наводнений вследствие изменения климата и экстремальных осадков, что требует развития систем раннего предупреждения, инженерной инфраструктуры и просветительской работы для населения [8]. В Польше озёра подверглись значительной деградации из-за поступления неочищенных сточных вод, нарушивших буферные свойства и химический состав [9]. Восстановление с применением методов последовательного коагулирования и связывания фосфора способствовало снижению эвтрофикации и улучшению качества воды, что подчеркивает важность комплексных технологических подходов и долгосрочного мониторинга для устойчивого управления водными экосистемами.

Особенно интересен случай естественного заболачивания осушенных водно-болотных угодий в районе Ольштынского озера в Польше [10], где в XIX веке проводились масштабные дренажные работы, исчезло более 140 озёр общей площадью около 3000 га. Однако восстановление ранее осушенных озёр остаётся недостаточно изученным.

Современные наблюдения фиксируют постепенное восстановление части этих водоёмов, в основном за счёт естественных процессов. Тем не менее, качество воды и трофический статус остаются нестабильными: повышенное содержание фосфора фиксируется в восстановленных озёрах и болотах, а отсутствие системных программ восстановления делает дальнейший прогресс зависимым от климатических условий.

Глобально эвтрофикация остаётся одной из главных проблем для пресных вод, вызывая цветение водорослей, дефицит кислорода, снижение биоразнообразия и угрозы для здоровья [11]. Эвтрофицированные осадки становятся источниками парниковых газов, усугубляя климатические изменения. Современные меры по восстановлению водоёмов включают снижение внешнего поступления питательных веществ и внутренние вмешательства – удаление биомассы и донных отложений с использованием их в сельском хозяйстве и энергетике, что способствует переходу к замкнутой экономике. Примером успешного технологического вмешательства является использование модифицированного бентонита с лантаном

(LMB) в озере Берензее (Германия), позволившего стабилизировать фосфор в донных отложениях и снизить биопродуктивность, предотвращая массовое цветение цианобактерий [12].

Похожий случай наблюдался в озере Фельдбергер Хаусзе (Северо-Восточная Германия), где длительный процесс восстановления после прекращения сброса сточных вод в 1980-х завершился переходом к мезотрофному состоянию лишь к 2015 году, чему способствовали как природные процессы, так и биоманипуляции, включая осаждение фосфора и регулирование зоопланктона [13].

Таким образом, международный опыт показывает, что сохранение и восстановление водных экосистем требуют комплексных, регионально адаптированных стратегий, сочетающих научный мониторинг, технологические инновации, эффективное управление ресурсами и вовлечение местного населения. Важна интеграция природных, социально-экономических и управленческих факторов для обеспечения экологической устойчивости водных ресурсов и повышения качества жизни населения.

В этих условиях авторы считают, что цифровые двойники водных экосистем могут стать эффективным инструментом сохранения и управления водными ресурсами.

В настоящее время применение концепции цифровых двойников в рамках природных экологических и экосистем набирает популярность в мире [14 – 16]. Например, предлагается создание инфраструктуры пространственных данных для цифрового двойника лесной экосистемы. Цифровой двойник EcoPro предназначен для прогнозирования состояния экосистем под воздействием климатических и антропогенных факторов, предоставляя доступ к модели «Earth System Model».

Цифровой двойник объекта (ЦД) – система, состоящая из цифровой модели объекта и двусторонних информационных связей с объектом (при наличии изделия) и (или) его составными частями.

Цифровые двойники функционируют в режиме реального времени параллельно с их физическими оригиналами, обеспечивая мониторинг состояния, анализ изменений и прогнозирование последствий различных сценариев. Это обеспечивает возможность оптимизации режимов эксплуатации водных ресурсов с учетом экологических ограничений регионального, федерального и мирового масштабов, что особенно важно в условиях изменения климата и роста антропогенной нагрузки. В отличие от традиционного цифрового моделирования, концепция цифровых двойников предполагает наличие динамической связи с объектом, постоянную актуализацию данных

и возможность использования результатов анализа для управления. Преимущества цифровых моделей заключаются в том, что они позволяют делать предсказания и сравнивать их с реальными данными. Однако такие модели учитывают лишь часть факторов, а сложность экспоненциально возрастает с увеличением числа переменных. Кроме того, недостатками являются отсутствие связи с физическим объектом в реальном времени и невозможность управления им. На рисунке 1 представлена обобщённая схема функционирования цифрового двойника.

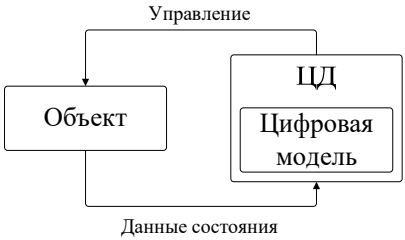


Рис. 1. Обобщённая схема функционирования цифрового двойника

Цифровые двойники можно классифицировать на три категории: цифровые двойники-прототипы (Digital Twin Prototype, DTP), экземпляры (Digital Twin Instance, DTI) и агрегированные двойники (Digital Twin Aggregate, DTA) [17].

DTP – это прототип некоторого физического объекта, включающий информационную модель, содержащую данные, необходимые для проектирования и последующего создания физического объекта, включая его конструктивные характеристики, технологические параметры и материалы. Такие модели широко используются в высокотехнологичных отраслях – от машиностроения до авиационной промышленности.

DTI – это двойник-экземпляр, который применяется для моделирования уже существующего физического объекта, к которому он привязан на протяжении всего жизненного цикла. Этот тип включает цифровое представление геометрии, параметров эксплуатации, истории обслуживания, а также результаты мониторинга и прогнозируемые изменения состояния.

DTA – это двойники-агрегаторы, в свою очередь, представляет собой интеграционную платформу, объединяющую множество DTI в рамках единой распределённой информационно-вычислительной системы, что открывает возможности для комплексного анализа

и управления группой объектов для последующего воздействия на их состояние.

В работе предлагается описание модульной архитектуры цифрового двойника водного объекта для обеспечения экологической устойчивости, включая мониторинг, прогнозирование и управление водными ресурсами в условиях антропогенного воздействия и климатических изменений.

2. Общее описание модульной архитектуры цифрового двойника водного объекта. Авторами предлагается следующая модульная архитектура цифрового двойника водного объекта, которая представлена замкнутой рекурсивной системой и включает следующие основные модули (рисунок 2): реальный водный объект (экосистема), сбор и слияния мультимодальных данных, цифровые модели экосистемы, экспертную систему, регулирование нормативно-правовой базы, геоинформационную систему (ГИС), графическое представление данных. Связь между модулями позволяет организовать обмен данными, обеспечивая оптимизацию управления природным объектом.

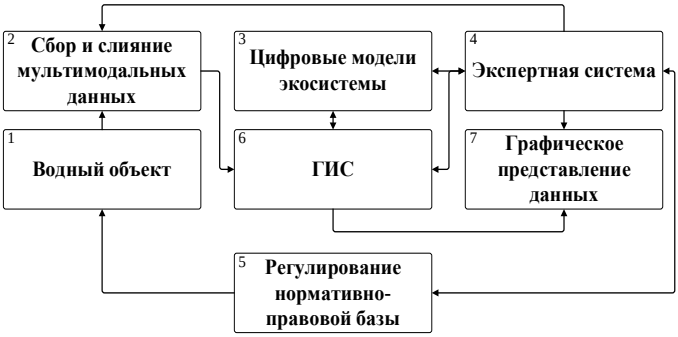


Рис. 2. Модульная архитектура цифрового двойника водного объекта

Следует отметить, что в настоящее время существуют информационные системы такие как GRDC, HydroSHEDS, USGS, «Метео ДВ», «Волга», «Гидрология», «Речные бассейны Европейской России», применяемые в сфере мониторинга и управления водными ресурсами [18 – 26]. Такие системы характеризуются отсутствием комплексной интеграции мультимодальных источников данных, недостаточной связанностью с прогнозными моделями экосистем, ограниченной реализацией экспертных модулей и слабой поддержкой механизмов регулирования состояния водных объектов в реальном времени. В результате указанные выше информационные системы

выполняют роль **неполных** цифровых двойников, позволяя лишь частично моделировать процессы в водных экосистемах. На рисунке 3 приведён модульный состав широко используемых неполных цифровых двойников.

Компоненты	1	2	3	4	5	6	7
Водный объект (данные наблюдений)	±	+	+	+	±	±	±
Сбор и слияние мультимодальных данных		±		±	±	±	
Модели экосистемы	±	±	±	±			
ГИС (геоинформационная составляющая)	+	+	+	+	+	+	+
Графическое представление данных	±	+	+	+	±	±	+
Экспертная система							
Регулирование нормативно-правовой базы (управляющие воздействия)							

Рис. 3. Модульный состав существующих информационных систем (неполных цифровых двойников): 1. GRDC; 2. HydroSHEDS; 3. USGS; 4. ГИС «Метео ДВ»; 5. ГИС «Волга»; 6. ГИС «Гидрология»; 7. Геопортал «Речные бассейны Европейской России»

Предлагаемая архитектура цифрового двойника в рамках исследования предполагает обязательное наличие контура управления объектом исследования – водным объектом (экосистемой водного объекта). Контур управления представляется в общем виде следующей информационной последовательностью: через модуль сбора и слияния мультимодальных данных, информация, переданная в геоинформационную систему, поступает в модуль цифровой модели экосистемы водного объекта и из нее в модуль экспертной системы к модулю регулирования нормативно-правовой базы и через него, за счет принимаемых решений обратно к водному объекту.

3. Модуль «Сбор и слияние мультимодальных данных». Важным компонентом предлагаемой системы является модуль сбора и слияния мультимодальных данных (рисунок 4), который интегрирует информацию из разнородных источников, включая спутники, наземные сенсоры, гидрологические и метеорологические станции, а также данные, поступающие из модуля экспертной системы нормативно-правовой базы функционирования природного объекта. Основные функциональные компоненты модуля представлены в таблице 1.

Ключевым компонентом эффективного функционирования цифрового двойника является обеспечение полноты, достоверности и актуальности данных. Для комплексного мониторинга водных ресурсов необходимо систематическое получение и интеграция информации о водном уровне, качестве воды, гидрологических характеристиках, метеорологических и климатических параметрах, а также о водопользовании. Кроме того, требуется учет данных, регулирующих состояние водных объектов в соответствии с установленными нормативами. Источниками информации могут быть: спутниковые системы дистанционного зондирования Земли; наземные гидрологические и метеостанции; локальные комплексные системы мониторинга; данные полевых выездов (аэрофотосъемка и данные собранные надводным аппаратом); данные из экспертной системы, включая нормативно-правовую базу функционирования природного объекта.

Таблица 1. Основные функциональные компоненты модуля «Сбор и слияние мультимодальных данных»

Класс / Метод	**Название процесса**	**Входные данные**	**Выходные данные**	**Исполнитель**
DataCollector. collect()	Сбор данных с датчиков, спутников, нормативных источников	Потоки данных	Сырые разнородные данные, временные метки	Цифровая система
DataMerger. merge()	Слияние мультимодальных данных	Сырые разнородные данные, временные метки	Единая база геопространственных данных (ГИС-хранилище)	Цифровая система
DataValidator. clean()	Автоматическая верификация и очистка, обеспечение совместимости с международными стандартами	Единая база геопространственных данных	Очищенные структурированные данные	Цифровая система
DataValidator.update()	Сохранение данных в базу данных	Очищенные структурированные данные	Обновленная единая база геопространственных данных (ГИС-хранилище)	Цифровая система

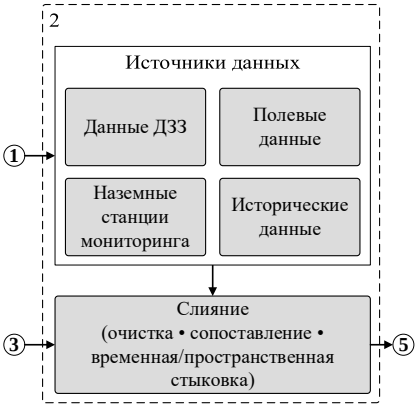


Рис. 4. Структура модуля «Сбор и слияние мультимодальных данных»

Частота обновления данных варьируется в зависимости от типа источника. Наземные станции могут передавать данные каждые несколько минут или часов, что позволяет поддерживать непрерывный мониторинг. Спутниковые данные обновляются в зависимости от периодичности пролета спутника над регионом, что обычно составляет от нескольких дней до недели. Полевые данные собираются с периодичностью, зависящей от специфики задач мониторинга – это могут быть как ежемесячные, так и сезонные замеры.

4. Модуль «Цифровые модели экосистемы». Цифровая модель объекта – система математических и компьютерных моделей, а также электронных документов объекта, описывающая структуру, функциональность и поведение вновь разрабатываемого или эксплуатируемого объекта на различных стадиях жизненного цикла, для которой на основании результатов цифровых и (или) иных испытаний выполнена оценка соответствия предъявляемым к объекту требованиям. Основу цифровой модели в рамках цифрового двойника водного объекта составляют математические, физические и вычислительные методы, которые позволяют воспроизводить динамику процессов, протекающих в природных экосистемах. Основные функциональные компоненты модуля представлены в таблице 2.

В общем случае, согласно [27], существующие математические методы моделирования водных экосистем в зависимости от объекта или процесса исследования подразделяются на гидрофизические, биологические и динамические модели (представляют собой интеграцию биологических и гидрофизических моделей).

Таблица 2. Основные функциональные компоненты модуля «Цифровые модели экосистемы»

Класс / Метод	Название процесса	Входные данные	Выходные данные	Исполнитель
EcoModel.train_model()	Построение / обучение цифровой модели машинного обучения	Обновленная единая база геопространственных данных	Настроенная цифровая модель машинного обучения	Цифровая система
EcoModel.simulate()	Прогноз состояния водного объекта	Настроенная модель машинного обучения + новые данные	Прогнозные показатели состояния экосистемы	Цифровая система
EcoModel.validate()	Верификация / калибровка	Прогноз + реальные данные	Откалиброванная модель	Цифровая система
EcoModel.PhysicSimulator.simulate_scenarios()	Сценарное моделирование биологических и гидрофизических моделей	Климатические, гидрологические, биологические данные из базы геопространственных данных	Прогноз динамики экосистемы, сценарии изменений	Цифровая система
EcoModel.Data.update()	Сохранение данных в базу данных	Результаты EcoModel.simulate() и EcoModel.EcosystemSimulator.simulate_scenarios()	Обновленная единая база геопространственных данных	Цифровая система
EcoModel.getData()	Получение данных цифровой модели и физического моделирования	Результаты EcoModel.simulate() и EcoModel.EcosystemSimulator.simulate_scenarios()	Сводные данные из EcoModel.simulate() и EcoModel.EcosystemSimulator.simulate_scenarios()	Цифровая система

Модели машинного обучения используются для оперативного прогноза интегральных показателей состояния водного объекта на основе обновляемой единой базы геопространственных данных. В частности, они применяются для кратко- и среднесрочного прогноза параметров качества воды и общего состояния экосистемы при поступлении новых данных, а также для последующей верификации и калибровки.

Физические и биофизические модели применяются в рамках сценарного моделирования и ориентированы на анализ причинно-следственных процессов и долгосрочных изменений экосистемы. Они используются для оценки динамики уровня воды, гидрологических и гидрофизических характеристик, а также биологических процессов, включая сценарии эвтрофикации и реакции экосистемы на климатические и антропогенные воздействия.

В зависимости от типа задач и доступных данных применяются различные методологии моделирования [28]: физическое моделирование, моделирование с помощью модели, управляемой данными, моделирование с помощью модели машинного обучения.

Физические модели основываются на законах природы и описывают процессы с использованием уравнений механики, гидродинамики, термодинамики и других разделов физики. Они особенно эффективны при наличии глубокого понимания процессов, происходящих в системе.

Модели, управляемые данными, строятся на основе статистического анализа и позволяют выявлять зависимости между параметрами объекта, основываясь на эмпирических данных. Такие модели часто используются для анализа сложных систем с большим количеством взаимосвязанных факторов.

Модели машинного обучения применяются для решения задач прогнозирования и классификации. Эти модели обучаются на исторических данных и используют алгоритмы глубокого обучения или другие методы искусственного интеллекта для моделирования поведения и определения ключевых закономерностей.

Данные, поступающие от реальных объектов, играют ключевую роль в построении и использовании цифровой модели. Они используются для: обучения моделей, что позволяет создавать точные представления природных объектов и их поведения; калибровки, при которой параметры модели подстраиваются для соответствия реальным условиям; верификации и валидации, что предполагает сравнение прогностических результатов модели с эмпирическими данными для повышения точности.

На сегодняшний день разработано множество систем для моделирования водных экосистем, которые активно применяются в научной и практической деятельности: AQUATOX, SWAT (Soil and Water Assessment Tool), CE-QUAL-W2, Ecopath, Loop Analysis и STELLA [29, 30].

5. Модуль «Экспертная система». Модуль «Экспертная система» (рисунок 5) представляет собой кибернетическую систему

поддержки принятия решений, функционирующую на основе формализованных знаний о динамике водного объекта и нормативно-правовых ограничениях. Экспертная система играет важную роль в анализе законодательной базы и консолидации данных для построения прогнозов и содержит базы данных нормативных актов, связанных с процессом природопользования, модуль для анализа документов, экспертную базу данных, модуль поддержки и принятия управленческих решений, систему генерации отчетности о состоянии природного объекта. Основные функциональные компоненты модуля представлены в таблице 3.

Таблица 3. Основные функциональные компоненты модуля «Экспертная система»

Класс / Метод	**Название процесса**	**Входные данные**	**Выходные данные**	**Исполнитель**
ExpertSystem.integrate_data()	Интеграция данных из моделей и ГИС	Сводные данные из EcoModel.getData() + экспертная база	Сводные данные для анализа	Цифровая системы
ExpertSystem.analyze_scenarios()	Многокрите-риальный анализ сценариев	Сводные данные ExpertSystem.integrate_data()	Оценка рисков и сценариев	Цифровая системы
ExpertSystem.generate_recommendations()	Генерация рекомендаций	Оценка сценариев	Рекомендации для регуляторов и природополь-зователей	Цифровая системы

Нормативно правовая база актов включает: федеральные и региональные законы, регулирующие природопользование; нормативы и стандарты качества воды и состояния окружающей среды; международные соглашения и экологические программы по улучшению состояния окружающей среды [31 –33].

Современные методы обработки текста играют ключевую роль в работе экспертной системы. Для анализа и извлечения информации из нормативно-правовых документов, а также для генерации отчетов используются алгоритмы машинного обучения, искусственные нейронные сети, в частности большие языковые модели [34]. Эти алгоритмы способны выделять ключевую информацию из больших объемов текстов и проводить семантический анализ, сопоставляя содержание документа с текущей ситуацией. В экспертной системе большие языковые модели используются в таких модулях как анализ

документов и системе генерации отчетности о состоянии природного объекта.

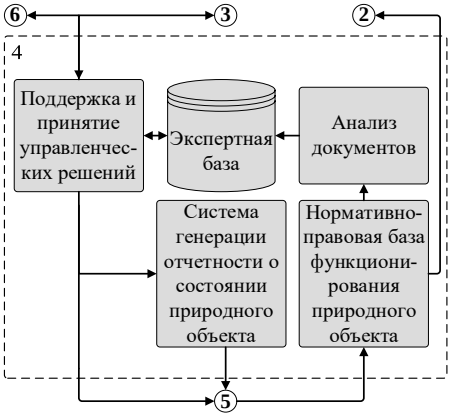


Рис. 5. Структура модуля «Экспертная система»

Ключевым свойством системы является ее адаптивность, обеспечиваемая механизмами непрерывного обновления базы знаний (как за счет поступления новых нормативных актов, так и через накопление прецедентов принятия решений) и динамической калибровки моделей на основе данных мониторинга. Это позволяет рассматривать модуль как интеллектуальный регулятор в контуре управления сложной эколого-технической системой, где объект управления (водный объект) характеризуется существенной нелинейностью, распределенными параметрами и стохастическими возмущениями. Теоретической основой функционирования системы выступает синтез принципов оптимального управления (критерии качества водной среды) и устойчивость к неопределенностям входных данных, что обеспечивает выполнение целевых показателей состояния экосистемы в условиях изменяющихся внешних воздействий. Математический аппарат системы включает: аппарат нечеткой логики для работы с неопределенностями в экспертных знаниях.

Особое внимание уделяется анализу документов. Модуль «Анализ документов» представляет собой автоматизированную систему обработки и семантического анализа нормативно-правовых актов [35], технической документации и регламентирующих материалов, связанных с управлением водными ресурсами. Его основная функция заключается в преобразовании неструктурированных текстовых данных в формализованные знания,

пригодные для интеграции в экспертную систему водного объекта. Основные функциональные компоненты модуля представлены в таблице 4.

Таблица 4. Основные компоненты модуля «Анализ документов»

Класс / Метод	Название процесса	Входные данные	Выходные данные	Исполнитель
DocAnalyzer.parse()	Сбор и конвертация документов	Государственные реестры, базы законодательства	Структурированные тексты	Цифровая система
DocAnalyzer.extract_entities()	Извечение нормативно-значимых сущностей	Структурированные тексты	Нормативные параметры и связи	Цифровая система
DocAnalyzer.integrate_with_expert_db()	Сохранение данных в экспертной БД	Нормативные параметры	Обновлённая экспертная база данных	Цифровая система

Модуль анализа документов реализует комплексный подход к обработке документов, включающий следующие ключевые процессы: автоматизированный сбор и предварительная обработка документов из разнородных источников (государственные реестры, базы законодательства, локальные хранилища); лингвистический анализ текстов с применением методов обработки естественного языка и машинного обучения; извлечение нормативно-значимых сущностей и параметров; выявление семантических связей и потенциальных противоречий между различными нормативными актами.

В основе работы модуля лежит многоуровневая архитектура обработки данных. На первом уровне осуществляется парсинг и конвертация документов из исходных форматов в структурированное представление. На втором уровне происходит предварительная обработка исходного текста с помощью методов токенизации и морфологического анализа. На третьем уровне применяются алгоритмы машинного обучения, включая трансформерные модели, для семантического анализа текста, распознавания именованных сущностей (выделение нормативов, организаций, географических объектов, дат) и классификации документов. Четвертый уровень отвечает за интеграцию извлеченных данных с экспертными знаниями и нормативными базами, включая механизмы разрешения конфликтов и верификации данных. Пятый уровень отвечает за хранение информации в экспертной базе и полнотекстовый поиск по ней.

Количество выпущенных нормативных и правовых актов, содержащихся в информационной системе, охватывающей все федеральные и ведомственные нормативно-правовые документы, по следующим ключевым запросам: «водные ресурсы», «водные экосистемы», «природные объекты» – составило 2798 единиц. Для дальнейшего анализа было отобрано 500 документов в формате PDF, содержащих наиболее релевантную информацию по теме управления водными объектами и охраны водной среды. Поскольку анализ выбранных документов представляет собой обработку большого объема неструктурированной текстовой информации, были рассмотрены различные подходы с использованием больших языковых моделей (LLM). Таблица 5 демонстрирует сравнительную эффективность различных подходов к реализации языковых моделей при решении задач анализа нормативно-правовой документации. Основным критерием оценки выступало качество ответов (% корректных или релевантных результатов) при работе с запросами, связанными с тематикой водных ресурсов и экосистем.

Таблица 5. Сравнение эффективности различных подходов в реализации языковых моделей

Модель	Качество ответов, %	Машинное обучение
llama3.1 + similarity_search	80	Трансформер
RAG llama3.1	70	Трансформер
deepseek-r1	60	Трансформер
YandexGPT 5 Pro	90	Трансформер
multilingual-e5-large(Эмбеддинг) + similarity_search	80	Трансформер(XLM-RoBERTa-large)

Наилучшие показатели продемонстрировала модель YandexGPT 5 Pro, обеспечив 90% качества ответов, что может быть связано с ее обучением на русскоязычных данных. Следует отметить, что YandexGPT 5 Pro – единственная модель из таблицы 5, которая функционировала в облачной среде с ограничением на обработку не более 25 документов за сессию, однако, при увеличении объема обрабатываемых документов точность этой модели потенциально может снизиться.

Также хорошие результаты показали подходы на основе llama3.1 с механизмом similarity search и multilingual-e5-large, достигнув 80% качества. Использование RAG-архитектуры с llama3.1 оказалось менее

эффективным (70%), что, вероятно, связано с ограничениями по полноте и релевантности retrieved-документов. Модель deepseek-r1 показала худший результат – 60%, что может свидетельствовать о недостаточной адаптации к правовой тематике или ограниченной поддержке русского языка. Возможные направления улучшения результатов:

– адаптация моделей к предметной области: дополнительное дообучение на корпусе нормативно-правовых актов, связанных с водными ресурсами и экологическим регулированием, может значительно повысить точность ответов;

– улучшение retrieval-компоненты в RAG: повышение качества поиска релевантных фрагментов, через обучение эмбеддингов – улучшение векторных представлений текста для задач поиска, кластеризации или классификации.

Таблица 6 представляет результаты сравнительного анализа трёх различных подходов к извлечению информации из 500 штук нормативно-правовых документов. Основным критерием эффективности выступает процент корректных извлечений информации по ключевым признакам (тематическая релевантность, точность, полнота).

Таблица 6. Сравнение эффективности LLM и ручного анализа

Метод	Корректность, %
RAG LLM	76
Регулярные выражения	49
Эксперт	98

Наибольшую точность продемонстрировал экспертный (ручной) анализ, обеспечив 98% корректности ответов. Это объясняется глубоким пониманием контекста, правовой терминологии и способности интерпретировать сложные формулировки, которые часто остаются за пределами возможностей формализованных алгоритмов. Однако основной недостаток экспертного подхода – низкая скорость обработки и невозможность масштабирования на большие массивы данных без существенных затрат.

Метод с использованием RAG LLM показал 76% корректности ответов, что делает его наиболее сбалансированным подходом с точки зрения соотношения точности и скорости.

Метод на основе регулярных выражений дал лишь 49% корректных результатов, что объясняется его жесткой структурой, неустойчивостью к вариативности юридического языка и

невозможностью учета контекста. Регулярные выражения хорошо работают только при строго повторяющихся шаблонах, но не подходят для обработки юридических текстов с множественными синтаксическими конструкциями и вариациями формулировок.

6. Модуль «Регулирование нормативно-правовой базы». Модуль «Регулирование нормативно-правовой базы» отвечает за управление водным объектом. В этом модуле результаты работы экспертной системы анализируются лицами, принимающими решения (органы власти и субъекты природопользования), при этом субъектом природопользования может выступать физический объект или цифровой двойник. Принимающие решения лица, опираясь на полученные данные, принимают решение о загрязненности водного объекта и вырабатывают дальнейшие действия по устранению причин загрязнений. Процесс управления водными объектами является циклическим и может быть представлен в виде структуры (рисунок 6). Основные функциональные компоненты модуля представлены в таблице 7.

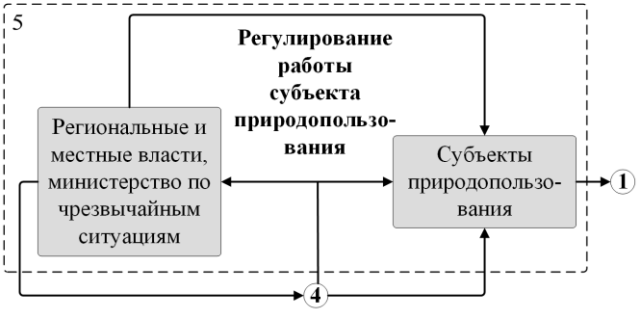


Рис. 6. Структура модуля «Регулирование нормативно-правовой базы»

В процессе принятия решений лицо, принимающее решение сталкивается с множеством разнородных, а зачастую и противоречивых мнений. В связи с этим утверждение о существовании единственного оптимального варианта действий далеко не всегда соответствует действительности.

Следует учитывать, что сама ситуация выбора может иметь многоуровневую структуру и делиться на отдельные взаимосвязанные подзадачи. Каждая из них может быть формализована с помощью специализированной оптимизационной цифровой модели. Таким образом, процесс принятия управленческого решения в сложных условиях можно рассматривать как последовательное решение ряда частных задач, выстроенных в определённой иерархии, а осуществлять управление состоянием природного

нужно объекта, в этом случае, за счет автоматической выработки управленческих решений в рамках цифрового двойника.

Таблица 7. Основные функциональные компоненты модуля «Регулирование нормативно-правовой базы»

Класс / Метод	**Название процесса**	**Входные данные**	**Выходные данные**	**Исполнитель**
RegulationModule.react_changes()	Механизм оперативного реагирования	Отчёты об изменениях законодатель-ства + рекомендации экспертной системы	Новые нормативные требования	Человек + Цифровая система
RegulationModule.update_database()	Формирование базы данных нормативных документов	Новы нормативные требования	Актуализи-рованная база нормативных актов	Человек (утверждает) + Цифровая система (загружает)
	Формирование предписаний и уведомлений субъектам природопользования	Утвержденные решения, адресаты	Предписания/ уведомления (PDF/API), сроки исполнения	Человек
	Учёт исполнения, сбор подтверждений и обратной связи	Предписания, отчеты исполнителей, телеметрия	Статусы исполнения, нарушения сроков	Человек
	Применение санкций/стимулов при невыполнении/ перевыполне-нии	Статус исполнения, регламент, санкции	Решение о мерах воздействия	Человек + Цифровая система

Исходя из этого, опираясь на результаты системного анализа, аналитических прогнозов и рекомендаций экспертной системы предлагается три алгоритма работы:

1. принятие управленческих решений происходит в полуавтоматическом режиме и контролируется ответственным лицом;
2. принятие управленческих решений контролируется ответственным лицом;
3. управленческое решение принимается ответственным лицом с помощью когнитивных и аналитических инструментов,

которые обеспечивают поддержку принятия решений и предоставляются цифровым двойником.

7. Описание работы цифрового двойника водного объекта по предложенной архитектуре. Исходя из описанных выше модулей архитектура цифрового двойника водного объекта представляется схемой на рисунке 7.

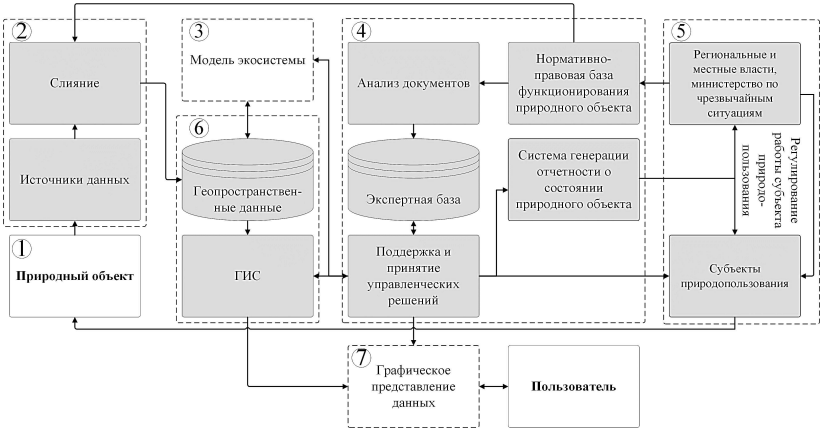


Рис. 7. Архитектура цифрового двойника водного объекта

Рисунок 7 представляет собой расширенную архитектуру цифрового двойника водного объекта с детализированными функциональными модулями и потоками данных, тогда как рисунок 2 демонстрирует укрупнённую структурную схему ключевых модулей без детальной проработки внутренних связей.

Описание принципа работы системы цифрового двойника выглядит следующим образом. В режиме реального времени система сбора и слияния мультимодальных данных фиксирует данные с датчиков мониторинга параметров экосистемы природного объекта в базе геопространственных данных, которая является частью геоинформационной системы. Источниками данных выступают данные с датчиков мониторинга качества воды, таких как измерители температуры, pH и концентрации загрязняющих веществ, а также устройства для отслеживания флоры и фауны природного объекта. Помимо этого, источниками данных являются метеорологические станции и системы дистанционного зондирования Земли, а также модуль экспертной системы нормативно-правовой базы функционирования природного объекта. Полученные данные составляют основу информационного наполнения системы.

Собранные данные передаются в цифровую модель экосистемы, где они обрабатываются и анализируются.

Данные экспертной системы, а также данные из цифровой модели экосистемы передаются в геоинформационную систему. В этой системе информация визуализируется и становится доступной для пространственного анализа. Цифровая модель экосистемы предоставляет экспертной системе необходимые данные для проведения анализа и выработки рекомендаций. Экспертная система использует эти данные вместе с нормативными актами и параметрами, хранящимися в базе геоинформационной системы, чтобы оценить текущее состояние природного объекта и предложить меры по его улучшению, рационализации использования или сохранению.

Из модуля анализа документов выделенная семантическая информация сохраняется в экспертной базе данных для дальнейшей работы модуля поддержки и принятия управленческих решений. На основе новых данных, полученных из цифровой модели и семантических данных из экспертной базы данных модуль поддержки и принятия управленческих решений корректирует внутреннюю работу, принимает решение и передает данные для генерации отчетности о возможных нарушениях в экологической обстановке.

Решение по результатам работы экспертной системы, через систему генерации отчетности направляются субъектам природопользования, таким как компании, занимающиеся добычей полезных ископаемых, сельским хозяйством, лесозаготовкой и другими видами деятельности, оказывающими влияние на природные объекты. Это позволяет им адаптировать свою деятельность под текущие условия и минимизировать негативное воздействие на окружающую среду. Субъекты природопользования и другие цифровые двойники, используют результаты работы цифрового двойника для повышения эффективности своих операций.

Органы управления, такие как Минприроды России, Росводресурсы, Росприроднадзор, Росгидромет, министерство по чрезвычайным ситуациям, а также региональные и местные власти, используют данные экспертной системы для принятия обоснованных решений. Эти решения касаются регулирования использования природных ресурсов, предотвращения экологических угроз и реализации природоохранных мероприятий. Органы управления вносят изменения в базу данных нормативных актов, отражающие новые законодательные инициативы и требования. Эти обновления затем используются экспертной системой при анализе и выработке рекомендаций. Решения, принятые органами управления оказывают

непосредственное воздействие на субъекты природопользования, обязывая их соблюдать установленные стандарты и ограничения.

Представленная архитектура цифрового двойника водного объекта основана на распределенной микросервисной архитектуре. Для программной реализации «цифрового двойника» был определён следующий технологический стек: языки программирования – python, Rust; для контейнеризации и оркестрации сервисов – технологии Docker и Kubernetes; для организации очередей задач между микросервисами – брокер RabbitMQ; для хранения геопространственных данных – СУБД PostgreSQL с расширением PostGIS; авторские библиотеки моделей машинного обучения для классификации и семантической сегментации участков местности, а так же RAG LLM llama3.1 с библиотеками сематического поиска и векторизации текста; для доступа к нормативно-правовым документам – API Гарант.

Создание цифрового двойника водного объекта позволяет оперативнее реагировать на изменения, происходящие в природном объекте. Для примера рассмотрим ситуацию с антропогенным воздействием человека на экосистемы озера Кабан, РФ [36, 37]. При реализации цифрового двойника требуется проследить во времени изменение объекта, в данном случае озера Кабан, а также принятые нормативно-правовые акты, регулирующие воздействие человека.

Исторически известно, что озеро Кабан в Казани начали загрязняться с конца XVII века из-за кожевенных и мыловаренных мастерских, а в XIX–XX веках усугубилось сбросами от заводов (ТЭЦ, химкомбинат, обувной комбинат «Спартак»), что привело к цветению водорослей, гибели экосистемы, накоплению сероводорода, тяжелых металлов. Начиная с 1981 гг. благодаря правительственным постановлениям и мерам санэпиднадзора началась «реанимация» с предупреждениями и штрафами предприятиям и уже через 3 года ситуация в озерах улучшилась. Мероприятия проводимые с 1981 по 1994 гг.: 1981-1984 гг.– прекращение сброса промышленных стоков; канализование и отведение части хозяйственно-фекальных стоков; 1982 г.– создание временной проточности; кратковременная аэрация; 1984-1987 гг. – изъятие донных отложений (Нижний Кабан); 1988-1994 гг. – изъятие донных отложений (Средний Кабан). В 2011 г. выявили 27 источников стоков на Среднем Кабане, установили фитоочистные каскады с растениями для естественной фильтрации, водные сады и очистные на Нижнем Кабане, что запустило самовосстановление бактериями при закрытии стоков. До конца проблема очистки озер Кабан не решена, но концепция набережных с эко-энергосистемой, экологические нормативы РТ и федеральные

санитарные нормы, привели к благоустройству 1,5 км берегов и снижению токсичности.

Обобщая изложенную выше информацию, можно представить состояния озера Кабан на временной оси. В соответствии с временными метками информация поступает на целевые модули «цифрового двойника» где происходит обработка информации. На рисунке 8 графически представлен процесс обработки информации с привязкой к функциональным модулям архитектуры системы, представленной на рисунке 7. Горизонтальная ось времени иллюстрирует ключевые исторические и современные периоды антропогенного воздействия и восстановления водного объекта: от начального этапа деградации до современного этапа цифрового мониторинга и прогнозирования с помощью «цифрового двойника».

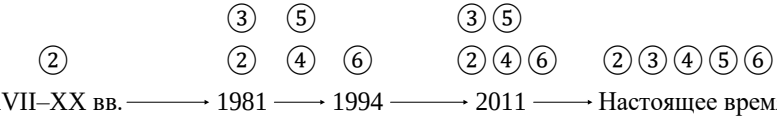


Рис. 8. Привязка модулей цифрового двойника водного объекта к о. Кабан

8. Заключение. Цифровой двойник экосистемы природного объекта в целом, и экосистемы водного объекта в частности, является передовым инструментом управления и контроля. Научная новизна работы заключается в разработке архитектуры цифрового двойника включающего обязательное наличие контура управления экосистемой водного объекта, учитывающего результаты работы экспертной системы, чего ранее не рассматривали существующие подходы. Интеграция методов машинного обучения, анализа больших данных и методов обработки естественного языка обеспечивает возможность выявления скрытых закономерностей, что значительно повышает качество управленческих решений. Предлагаемая архитектура цифрового двойника водного объекта позволяет оперативно реагировать на изменения, оптимизировать процесс природопользования и предотвращать экологические кризисы. Использование такой архитектуры создаёт основу для устойчивого природопользования, позволяя минимизировать негативные воздействия, оперативно реагировать на отклонения от экологических нормативов и обеспечивать долгосрочную сохранность водных объектов.

Литература

1. Jones K.R., Venter O., Fuller R.A., Allan J.R., Maxwell S.L., Negret P.J., Watson J.E. One-third of global protected land is under intense human pressure // Science. 2018. vol. 360. no. 6390. pp. 788–791. DOI: 10.1126/science.aap9565.
2. Liu M., Nie Z., Cao L., Wang L., Lu H. Nature-Based Solutions for the Restoration of Groundwater Level and Groundwater-Dependent Ecosystems in a Typical Inland Region in China // Water. 2024. vol. 16. no. 1. DOI: 10.3390/w16010033.
3. Bao N., Song W., Ma J., Chu Y. Multi-Source Remote Sensing Analysis of Yilong Lake's Surface Water Dynamics (1965–2022): A Temporal and Spatial Investigation // Water. 2024. vol. 16. no. 14. DOI: 10.3390/w16142058.
4. Chen M., Zhang R., Jia M., Cheng L., Zhao C., Li H., Wang Z. Accurate and rapid extraction of aquatic vegetation in the China side of the Amur River Basin based on Landsat imagery // Remote Sensing. 2024. vol. 16. no. 4. DOI: 10.3390/rs16040654.
5. Chao F., Jiang X., Wang X., Lu B., Liu J., Xia P. Water level fluctuation rather than eutrophication induced the extinction of submerged plants in Guizhou's Caohai Lake: Implications for lake management // Water. 2024. vol. 16. no. 5. DOI: 10.3390/w16050772.
6. Li Y., Tong J., Wang L. Full implementation of the river chief system in China: Outcome and weakness // Sustainability. 2020. vol. 12. no. 9. DOI: 10.3390/su12093754.
7. Li H., Jin L., Si Y., Mu J., Liu Z., Liu C., Zhang Y. Lake Restoration Improved Ecosystem Maturity Through Regime Shifts – A Case Study of Lake Baiyangdian, China // Sustainability. 2024. vol. 16. no. 21. DOI: 10.3390/su16219372.
8. Aksoy B. Flood Analysis in Lower Filyos Basin Using HEC-RAS and HEC-HMS Software // Sustainability. 2025. vol. 17. no. 11. DOI: 10.3390/su17115220.
9. Grochowska J.K., Gozdziejewska A.M., Augustyniak-Tunowska R. Changes in the Buffer Properties of the Restored Lake Complex // Sustainability. 2024. vol. 16. no. 18. DOI: 10.3390/su16187990.
10. Skwierawski A. Contemporary Evolution and Water Quality of Lakes Rewetted After 19th Century Drainage in the Olsztyn Lake District (Poland) // Water. 2024. vol. 16. no. 24. DOI: 10.3390/w16243661.
11. Tammeorg O., Chorus I., Spears B., Noges P., Nurnberg G.K., Tammeorg P., Sondergaard M., Jeppesen E., Paerl H., Huser B., Horppila J., et al. Sustainable lake restoration: From challenges to solutions // Wiley Interdisciplinary Reviews: Water. 2024. vol. 11. no. 2. pp. DOI: 10.1002/wat2.1689.
12. Epe T.S., Finsterle K., Yasseri S. Nine years of phosphorus management with lanthanum modified bentonite (Phoslock) in a eutrophic, shallow swimming lake in Germany // Lake and Reservoir Management. 2017. vol. 33. no. 2. pp. 119–129. DOI: 10.1080/10402381.2016.1263693.
13. Kasprzak P., Gonsiorczyk T., Grossart H.P., Hupfer M., Koschel R., Petzoldt T., Wauer G. Restoration of a eutrophic hard-water lake by applying an optimised dosage of poly-aluminium chloride (PAC) // Limnologica. 2018. vol. 70. pp. 33–48. DOI: 10.1016/j.limno.2018.04.002.
14. Шайтура С.В., Митрофанов Е.М., Жаров В.Г., Феоктистова В.М. Инфраструктура пространственных данных для цифрового двойника лесной экосистемы // Дизайн и технологии. 2022. № 91-92(133-134). С. 160–168.
15. Lee S., Kalmus P., Ferraz A., Goodman A., Pearson K., Doran G., Platt F., Hu B., Ekanayaka A., Chakraborty S. Ecopro: Ecological Projection Digital Twin // IGARSS 2024 – IEEE International Geoscience and Remote Sensing Symposium. 2024. pp. 2311–1214.
16. Hu D., Zhou C., Xie F. Application of Digital Twin Technology in Assessing the Level of Water Ecological Civilization Construction in Yangtze River Basin // Hydraulic

Structure and Hydrodynamics. 2024. pp. 343–352. DOI: 10.1007/978-981-97-7251-3_30.
17. Потапов В.П., Кузьмин Д.Г., Сероус Т.О. Научно-практические основы проекта «Цифровой Ускат» и особенности его реализации // Уголь. 2022. № 11(1161). С. 40–47. DOI: 10.18796/0041-5790-2022-11-40-47.
18. Global Runoff Data Centre. URL: https://portal.grdc.bafg.de/applications/public.html?publicuser=PublicUser#dataDownload/Home (дата обращения: 25.09.2025).
19. HydroSHEDS. URL: https://www.hydrosheds.org/ (дата обращения: 25.09.2025).
20. USGS National Water Dashboard. URL: https://dashboard.waterdata.usgs.gov/app/nwd/en/?aoi=default (дата обращения: 25.09.2025).
21. ФГИС «Экомониторинг». URL: https://rfi.mnr.gov.ru/projects/247/ (дата обращения: 25.09.2025).
22. Frolov A.V., Asmus V.V., Borshch S.V., Vil'fand R.M., Zhabina I.I., Zatyagalova V.V., Krovotyntsev V.A., Kudryavtseva O.I., Leont'eva E.A., Simonov Y.A., Stepanov Y.A. GIS-Amur system of flood monitoring, forecasting, and early warning // Russian Meteorology and Hydrology. 2016. vol. 41. pp. 157–169. DOI: 10.3103/S1068373916030018.
23. GIS «Meteo DV» URL: https://meteo-dv.ru/ (дата обращения: 25.09.2025).
24. Дерюгина В.В., Василенко Е.В., Кровотынцев В.А., Кухарский А.В. Технологические решения при создании спутникового сегмента информационной системы мониторинга наводнений в бассейне р. Волга // Интерэкспо Гео-Сибирь. 2022. Т. 4. С. 40–47. DOI: 10.33764/2618-981X-2022-4-40-47.
25. Дерюгина В.В., Борщ С.В., Кровотынцев В.А., Симонов Ю.А. ВЕБ ГИС системы мониторинга и прогнозирования гидрологической обстановки в бассейнах крупных рек России с использованием наземных и спутниковых данных // Материалы 17-й Всероссийской открытой конференции «Современные проблемы дистанционного зондирования Земли из космоса» (г. Москва, 11–15 ноября 2019). М.: ИКИ РАН, 2019. С. 84.
26. Фролова Н.Л., Самсонов Т.Е., Киреева М.Б., Семин В.Н., Эн-тин А.Л. Веб-приложение «Водный режим рек Европейской территории России»: структура и функциональные возможности // Водное хозяйство России: проблемы, технологии, управление. 2024. № 2. С. 6–22. DOI: 10.35567/19994508-2024-2-6-22.
27. Щепетова В.А. Основы математического моделирования в экологии: монография. Пенза: ПГУАС. 2015. 122 с.
28. Dihan M.S., Akash A.I., Tasneem Z., Das P., Das S.K., Islam M.R., Islam M.M., Badal F.R., Ali M.F., Ahamed M.H., et al. Digital twin: Data exploration, architecture, implementation and future // Heliyon. 2024. vol. 10. no. 5. DOI: 10.1016/j.heliyon.2024.e26503.
29. Patonai K., Fabian V.A. Comparison of three modelling frameworks for aquatic ecosystems: practical aspects and applicability // Community Ecology. 2022. vol. 23. no. 3. pp. 439–451. DOI: 10.1007/s42974-022-00117-3.
30. Bai J., Zhao J., Zhang Z., Tian Z. Assessment and a review of research on surface water quality modeling // Ecological Modelling. 2022. vol. 466. DOI: 10.1016/j.ecolmodel.2022.109888.
31. Воеводина Л.А., Воеводин О.В. О необходимости внесения изменений в водный комплекс РФ // Экология и водное хозяйство. 2021. Т. 3. № 2. С. 99–112. DOI: 10.31774/2658-7890-2021-3-2-99-112.

32. Josefsson H. The Water Framework Directive and Transnational Situations: A Call to Rethink Transboundary Water Management? // Journal for European Environmental & Planning Law. 2025. vol. 22. no. 1-2. pp. 168–186. DOI: 10.1163/18760104-22010209.
33. Biswas S., Dandapat B., Alam A., Satpati L. India's achievement towards sustainable Development Goal 6 (Ensure availability and sustainable management of water and sanitation for all) in the 2030 Agenda // BMC Public Health. 2022. vol. 22. no. 1. DOI: 10.1186/s12889-022-14316-0.
34. Матвеева А.Р., Антонов Е.В. Методика формирования базы данных характеристик сложного технологического объекта с использованием больших языковых моделей. Вестник НИЯУ МИФИ. 2024. Т. 13(5). pp. 350–357. DOI: 10.26583/vestnik.2024.5.7.
35. Бирюкова Е.Д., Миронов А.С. Методы подготовки данных для алгоритмов классификации текста // Информационные технологии XXI века: сборник научных трудов. Хабаровск: Тихоокеанский государственный университет, 2023. С. 180–185.
36. Фролова Л.Л., Мингазова Н.М. Сравнительный анализ выбора восстановительных технологий для озер Кабан г. Казани // Вестник Тамбовского университета, сер.: Естественные и технические науки. 1996. Т. 1. № 2. С. 166–168.
37. Meeting on the elimination of industrial wastewater releases into Sredniy Kaban Lake. URL: https://eco.tatarstan.ru/eng/index.htm/news/127082.htm?ysclid=mjka2htb45872248994 (дата обращения: 10.12.2025).

Миронов Андрей Сергеевич — канд. техн. наук, доцент высшей школы, высшая школа кибернетики и цифровых технологий, Тихоокеанский государственный университет (ТОГУ). Область научных интересов: подводная робототехника, обработка сигналов, аппаратные и программные средства определения координат в морской среде, разработка цифровых двойников природных объектов и экосистем, алгоритмы обработки информации с использованием методов машинного обучения, проектирование навигационных систем и систем связи. Число научных публикаций — 100. andrei.s.mironov@yandex.ru; Тихоокеанская улица, 136, 680035, Хабаровск, Россия; р.т.: +7(421)222-4356.

Саенко Александр Александрович — аспирант, высшая школа кибернетики и цифровых технологий, Тихоокеанский государственный университет (ТОГУ). Область научных интересов: разработка цифровых двойников природных объектов, мультисенсорные системы сбора и обработки измерительной информации, методы повышения эффективности компьютерного зрения для гидроакустических систем. Число научных публикаций — 8. saa2410@mail.ru; Тихоокеанская улица, 136, 680035, Хабаровск, Россия; р.т.: +7(924)105-9678.

Фомина Екатерина Сергеевна — старший преподаватель высшей школы, высшая школа кибернетики и цифровых технологий, Тихоокеанский государственный университет (ТОГУ). Область научных интересов: программные средства определения координат в морской среде, методы и средства обработки информации, получаемой с мультисенсорных систем технического зрения, включая применение методов машинного обучения и системный анализ. Число научных публикаций — 51. fominaekt@gmail.com; Тихоокеанская улица, 136, 680035, Хабаровск, Россия; р.т.: +7(999)082-8010.

Поддержка исследований. Исследование выполнено за счет гранта Российского научного фонда №24-11-20024 «Методы и средства построения цифровых моделей русла, береговой линии и водоохранной зоны реки Амур и ее основных притоков».

DOI 10.15622/ia.25.2.3

A. MIRONOV, A. SAENKO, E. FOMINA

DEVELOPMENT OF A DIGITAL TWIN ARCHITECTURE FOR AN AQUATIC ECOSYSTEM TO PRESERVE ENVIRONMENTAL SUSTAINABILITY

Mironov A., Saenko A., Fomina E. **Development of a Digital Twin Architecture for an Aquatic Ecosystem to Preserve Environmental Sustainability.**

Abstract. The purpose of this work is to develop a digital twin architecture of an aquatic ecosystem to preserve environmental sustainability, including monitoring, forecasting, and management of water resources in the context of anthropogenic impact and climate change. The paper proposes a modular digital twin architecture that includes a module for collecting and combining multimodal data, digital ecosystem models, an expert system, a regulatory framework module, a geographic information system, and a graphical data representation module. The architecture is based on the integration of data from sensors, satellites, weather stations, and remote sensing systems, using machine learning models and large language models to analyze regulatory documentation. The developed architecture of the digital twin of a water body allows monitoring ecosystem parameters in real time, assessing compliance with regulatory requirements, and generating reports and recommendations for organizations involved in natural resource management and governing bodies. The effectiveness of various models of legal text processing was tested, and optimal approaches to their semantic analysis were determined. The expediency of integrating the digital twin into the state environmental management system is substantiated. The digital twin architecture proposed by the authors is a comprehensive tool for sustainable water resources management that allows predicting the state of water bodies, timely identifying risks and developing scientifically sound management solutions, thereby reducing environmental threats and preserving water resources.

Keywords: digital twin, water resources, aquatic ecosystems, natural resource management, expert system.

References

1. Jones K.R., Venter O., Fuller R.A., Allan J.R., Maxwell S.L., Negret P.J., Watson J.E. One-third of global protected land is under intense human pressure. Science. 2018. vol. 360. no. 6390. pp. 788–791. DOI: 10.1126/science.aap9565.
2. Liu M., Nie Z., Cao L., Wang L., Lu H. Nature-Based Solutions for the Restoration of Groundwater Level and Groundwater-Dependent Ecosystems in a Typical Inland Region in China. Water. 2024. vol. 16. no. 1. DOI: 10.3390/w16010033.
3. Bao N., Song W., Ma J., Chu Y. Multi-Source Remote Sensing Analysis of Yilong Lake's Surface Water Dynamics (1965–2022): A Temporal and Spatial Investigation. Water. 2024. vol. 16. no. 14. DOI: 10.3390/w16142058.
4. Chen M., Zhang R., Jia M., Cheng L., Zhao C., Li H., Wang Z. Accurate and rapid extraction of aquatic vegetation in the China side of the Amur River Basin based on Landsat imagery. Remote Sensing. 2024. vol. 16. no. 4. DOI: 10.3390/rs16040654.
5. Chao F., Jiang X., Wang X., Lu B., Liu J., Xia P. Water level fluctuation rather than eutrophication induced the extinction of submerged plants in Guizhou's Caohai Lake: Implications for lake management. Water. 2024. vol. 16. no. 5. DOI: 10.3390/w16050772.
6. Li Y., Tong J., Wang L. Full implementation of the river chief system in China: Outcome and weakness. Sustainability. 2020. vol. 12. no. 9. DOI: 10.3390/su12093754.

7. Li H., Jin L., Si Y., Mu J., Liu Z., Liu C., Zhang Y. Lake Restoration Improved Ecosystem Maturity Through Regime Shifts – A Case Study of Lake Baiyangdian, China. Sustainability. 2024. vol. 16. no. 21. DOI: 10.3390/su16219372.
8. Aksoy B. Flood Analysis in Lower Filyos Basin Using HEC-RAS and HEC-HMS Software. Sustainability. 2025. vol. 17. no. 11. DOI: 10.3390/su17115220.
9. Grochowska J.K., Gozdziejewska A.M., Augustyniak-Tunowska R. Changes in the Buffer Properties of the Restored Lake Complex. Sustainability. 2024. vol. 16. no. 18. DOI: 10.3390/su16187990.
10. Skwierawski A. Contemporary Evolution and Water Quality of Lakes Rewetted After 19th Century Drainage in the Olsztyn Lake District (Poland). Water. 2024. vol. 16. no. 24. DOI: 10.3390/w16243661.
11. Tammeorg O., Chorus I., Spears B., Noges P., Nurnberg G.K., Tammeorg P., Sondergaard M., Jeppesen E., Paerl H., Huser B., Horppila J., et al. Sustainable lake restoration: From challenges to solutions. Wiley Interdisciplinary Reviews: Water. 2024. vol. 11. no. 2. pp. DOI: 10.1002/wat2.1689.
12. Epe T.S., Finsterle K., Yasseri S. Nine years of phosphorus management with lanthanum modified bentonite (Phoslock) in a eutrophic, shallow swimming lake in Germany. Lake and Reservoir Management. 2017. vol. 33. no. 2. pp. 119–129. DOI: 10.1080/10402381.2016.1263693.
13. Kasprzak P., Gonsiorczyk T., Grossart H.P., Hupfer M., Koschel R., Petzoldt T., Wauer G. Restoration of a eutrophic hard-water lake by applying an optimised dosage of poly-aluminium chloride (PAC). Limnologica. 2018. vol. 70. pp. 33–48. DOI: 10.1016/j.limno.2018.04.002.
14. Shaitura S.V., Mitrofanov E.M., Zharov V.G., Feoktistova V.M. [Spatial data infrastructure for forest ecosystem digital twin]. Dizajn i texnologii – Design and technology. 2022. no. 91-92(133-134). pp. 160–168. (In Russ.).
15. Lee S., Kalmus P., Ferraz A., Goodman A., Pearson K., Doran G., Platt F., Hu B., Ekanayaka A., Chakraborty S. Ecopro: Ecological Projection Digital Twin. IGARSS 2024 – IEEE International Geoscience and Remote Sensing Symposium. 2024. pp. 2311–1214.
16. Hu D., Zhou C., Xie F. Application of Digital Twin Technology in Assessing the Level of Water Ecological Civilization Construction in Yangtze River Basin. Hydraulic Structure and Hydrodynamics. 2024. pp. 343–352. DOI: 10.1007/978-981-97-7251-3_30.
17. Potapov V. P., Kuzmin D. G., Serous T.O. [Scientific and practical foundations of the Digital Uskat Project and specific features of its implementation]. Ugol' – Coal. 2022. no. 11(1161). pp. 40–47. (In Russ.). DOI: 10.18796/0041-5790-2022-11-40-47.
18. Global Runoff Data Centre. Available at: https://portal.grdc.bafg.de/applications/public.html?publicuser=PublicUser#dataDownload/Home. (accessed 25.09.2025).
19. HydroSHEDS. Available at: https://www.hydrosheds.org/. (accessed 25.09.2025).
20. USGS National Water Dashboard. Available at: https://dashboard.waterdata.usgs.gov/app/nwd/en/?aoi=default. (accessed 25.09.2025).
21. FGIS «E'komonitoring» [FGIS «Ecomonitoring»]. Available at: https://rfi.mnr.gov.ru/projects/247/. (accessed 25.09.2025).
22. Frolov A.V., Asmus V.V., Borshch S.V., Vil'fand R.M., Zhabina I.I., Zatyagalova V.V., Krovotyntsev V.A., Kudryavtseva O.I., Leont'eva E.A., Simonov Y.A., Stepanov Y.A. GIS-Amur system of flood monitoring, forecasting, and early warning. Russian Meteorology and Hydrology. 2016. vol. 41. pp. 157–169. DOI: 10.3103/S1068373916030018.
23. GIS «Meteo DV». Available at: https://meteo-dv.ru/. (accessed 25.09.2025).

24. Deryugina V.V., Vasilenko E.V., Krovotintsev V.A., Kukharsky A.V. [Technological solutions used of the development segment of the flood monitoring information system in the Volga river basin]. Intere'kspo Geo-Sibir' – Interexpo Geo-Siberia. 2022. vol. 4. pp. 40–47. DOI: 10.33764/2618-981X-2022-4-40-47. (In Russ.).
25. Deryugina V.V., Borshch S.V., Krovotyntsev V.A. Simonov Yu.A. [WEB GIS system for monitoring and forecasting the hydrological situation in the basins of large rivers in Russia using ground and satellite data] Materialy 17-j Vserossijskoj otkrytoj konferencii «Sovremennye problemy distancionnogo zondirovaniya Zemli iz kosmosa» [Proceedings of the 17th All-Russian open conference Modern Problems of Remote Sensing of the Earth from Space]. Moscow: IKI RAN. 2019. pp. 84. (In Russ.).
26. Frolova N.L., Samsonov T.E., Kireyeva M.V., Semin V.N., Entin A.L. [WEB application «Water Regime of the Rivers of the European Part of Russia»: Structure and functionality]. Vodnoe xozyajstvo Rossii: problemy, texnologii, upravlenie – Water Sector of Russia: Problems, Technologies, Management. 2024. no. 2. pp. 6–22. DOI: 10.35567/19994508-2024-2-6-22. (In Russ.).
27. Shhepetova V.A. Osnovy matematicheskogo modelirovaniya v e'kologii: monografiya. [Fundamentals of mathematical modeling in ecology: monograph]. Penza: PSUAC. 2015. 122 p. (In Russ.).
28. Dihan M.S., Akash A.I., Tasneem Z., Das P., Das S.K., Islam M.R., Islam M.M., Badal F.R., Ali M.F., Ahamed M.H., et al. Digital twin: Data exploration, architecture, implementation and future. Heliyon. 2024. vol. 10. no. 5. DOI: 10.1016/j.heliyon.2024.e26503.
29. Patonai K., Fabian V.A. Comparison of three modelling frameworks for aquatic ecosystems: practical aspects and applicability. Community Ecology. 2022. vol. 23. no. 3. pp. 439–451. DOI: 10.1007/s42974-022-00117-3.
30. Bai J., Zhao J., Zhang Z., Tian Z. Assessment and a review of research on surface water quality modeling. Ecological Modelling. 2022. vol. 466. DOI: 10.1016/j.ecolmodel.2022.109888.
31. Voevodina L.A., Voevodin O.V. [On the need to making amendments to the water code of the Russian Federation]. E'kologiya i vodnoe xozyajstvo – Ecology and water management. DOI: 10.31774/2658-7890-2021-3-2-99-112. (In Russ.).
32. Josefsson H. The Water Framework Directive and Transnational Situations: A Call to Rethink Transboundary Water Management? Journal for European Environmental & Planning Law. 2025. vol. 22. no. 1-2. pp. 168–186. DOI: 10.1163/18760104-22010209.
33. Biswas S., Dandapat B., Alam A., Satpati L. India's achievement towards sustainable Development Goal 6 (Ensure availability and sustainable management of water and sanitation for all) in the 2030 Agenda. BMC Public Health. 2022. vol. 22. no. 1. DOI: 10.1186/s12889-022-14316-0.
34. Matveeva A.R, Antonov E.V. [Methodology for forming a database of characteristics of a complex technological object using large language models]. Vestnik Natsional'nogo Issledovatel'skogo Yadernogo Universiteta «MIFI». 2024. vol. 13. no. 5. pp. 350–357. (In Russ.).
35. Biryukova E.D., Mironov A.S. [Methods for data preparation for text classification algorithms]. Informacionnye texnologii XXI veka: sbornik nauchnyx trudov [Technologies of the XXI Century: Collection of Scientific Papers]. Khabarovsk: Pacific National University. 2023. pp. 180–185. (In Russ.).
36. Frolova L.L., Mingazova N.M. [Comparative analysis of restoration technologies to be chosen for lakes Kaban of Kazan] Vestnik Tambovskogo universiteta. Serija Estestvennye i tehnicheskie nauki – Tambov university reports. Series: Natural and technical sciences. 1996. vol. 1. no. 2. pp. 166–168. (In Russ.).

37. Meeting on the elimination of industrial wastewater releases into Sredniy Kaban Lake. Available at: https://eco.tatarstan.ru/eng/index.htm/news/127082.htm?ysclid=mjka2htb45872248994 (accessed 10.12.2025).

Mironov Andrey — Ph.D., Associate professor at the Higher School, Higher School of Cybernetics and Digital Technologies, Pacific National University (PNU). Research interests: underwater robotics, signal processing, hardware and software for determining coordinates in the marine environment, development of digital twins of natural objects and ecosystems, information processing algorithms using machine learning methods, design of navigation and communication systems. The number of publications — 100. andrei.s.mironov@yandex.ru; 136, Tikhookeanskaya St., 680035, Khabarovsk, Russia; office phone: +7(421)222-4356.

Saenko Aleksandr — Ph.D. student, Higher School of Cybernetics and Digital Technologies, Pacific National University (PNU). Research interests: development of digital twins of natural objects, multisensory systems for collecting and processing measurement information, methods for improving the efficiency of computer vision for hydroacoustic systems. The number of publications — 8. saa2410@mail.ru; 136, Tikhookeanskaya St., 680035, Khabarovsk, Russia; office phone: +7(924)105-9678.

Fomina Ekaterina — Senior lecturer at the Higher School, Higher School of Cybernetics and Digital Technologies, Pacific National University (PNU). Research interests: software tools for determining coordinates in the marine environment, methods and tools for processing information obtained from multisensor technical vision systems, including the application of machine learning methods and system analysis. The number of publications — 51. fominaekt@gmail.com; 136, Tikhookeanskaya St., 680035, Khabarovsk, Russia; office phone: +7(999)082-8010.

Acknowledgements. The research was supported by the Russian Science Foundation (grant No. 24-11-20024).

УДК 004.95 **DOI** 10.15622/ia.25.2.4

А.В. Воробьев, Г.Р. Воробьева

КОНТЕКСТНО-ЗАВИСИМЫЙ МЕТОД АДАПТИВНОЙ НАСТРОЙКИ ПАРАМЕТРОВ АВТОРЕГРЕССИОННЫХ МОДЕЛЕЙ ДЛЯ НЕСТАЦИОНАРНЫХ ВРЕМЕННЫХ РЯДОВ

Воробьев А.В., Воробьева Г.Р. **Контекстно-зависимый метод адаптивной настройки параметров авторегрессионных моделей для нестационарных временных рядов.**

Аннотация. Предлагается метод контекстно-зависимой настройки параметров авторегрессионных моделей для восстановления пропусков в нестационарных временных рядах. Ключевая особенность метода заключается в адаптивном выборе параметров модели ARIMA (p, d, q) на основе двух факторов контекста: длительности пропуска и уровня внешних возмущений в соответствующий период. В отличие от стандартных подходов автоматического подбора, ориентированных на глобальную оптимизацию для прогнозирования, разработанный алгоритм сужает пространство поиска моделей и осуществляет выбор оптимальной конфигурации с помощью локальной кросс-валидации, что позволяет учитывать специфические условия в области пропуска. Метод реализован в виде программного модуля на языке Python с модульной архитектурой, обеспечивающей вычислительную эффективность за счет кеширования и параллельных вычислений. Эффективность метода проверена в ходе эксперимента на реальных геомагнитных данных (компонента DBE_NEZ обсерватории Ловозеро). Результаты демонстрируют, что в условиях спокойной и слабовозмущенной геомагнитной обстановки (индекс SME = 50–200 нТл) метод обеспечивает высокую точность восстановления (R^2 = 0.71–0.85) для пропусков длиной от 5 до 120 минут. При этом показано, что точность закономерно снижается с ростом уровня возмущений, что отражает фундаментальное ограничение, связанное с возрастающей стохастичностью исходного сигнала. Предложенный подход обеспечивает интерпретируемость и адаптивность, открывая перспективы для создания инструментов восстановления данных в различных прикладных областях.

Ключевые слова: авторегрессионные модели, восстановление пропусков, нестационарные временные ряды, геомагнитные данные.

1. Введение. Автоматический анализ и обработка нестационарных временных рядов представляет собой фундаментальную задачу в области информатики, машинного обучения и прикладной математики. Особую сложность в этом контексте вызывает проблема восстановления пропусков в данных, которые неизбежно возникают в реальных системах мониторинга вследствие сбоев оборудования, потери сигнала или иных технологических причин. Традиционным и хорошо зарекомендовавшим себя аппаратом для работы с такими рядами являются авторегрессионные модели, в частности, класс ARIMA-моделей, который за счет операции дифференцирования явно предназначен для анализа нестационарных процессов [1]. Однако эффективность этих моделей для задач интерполяции критически зависит от корректного выбора их

параметров – порядков авторегрессии (p), дифференцирования (d) и скользящего среднего (q) в модели ARIMA (p, d, q).

Существующие стандартные методы автоматического подбора параметров, такие как широко используемый алгоритм, реализованный в пакете «forecast» для R [2], ориентированы, в первую очередь, на задачи прогнозирования и основаны на минимизации информационных критериев (Акаике, Байеса) и анализе автокорреляционных функций. Несмотря на свою распространенность, эти методы обладают существенным ограничением при использовании для восстановления данных: они осуществляют выбор параметров, ориентируясь исключительно на внутреннюю статистическую структуру всего временного ряда, игнорируя локальные условия вокруг пропуска. В реальных же условиях, особенно при работе с данными физического мониторинга, финансовой аналитики или телеметрии, характер ряда в области пропуска может существенно зависеть от внешних, зачастую измеримых факторов. Длительный пропуск, возникший в период высокой внешней нагрузки на систему, принципиально отличается от пропуска аналогичной длины в период ее спокойной работы. Применение стандартных параметров, оптимизированных для глобального прогноза, к локальной задаче интерполяции может приводить к значительным погрешностям восстановления.

Таким образом, актуальной научно-технической задачей является разработка методов настройки моделей восстановления данных, которые бы учитывали не только внутренние глобальные закономерности ряда, но и локальный контекст, в котором наблюдаются пропуски. Подобный контекстно-зависимый подход позволяет перейти от универсальных, но зачастую излишне общих решений к адаптивным, которые тонко подстраиваются под конкретные условия восстанавливаемого сегмента данных. Это особенно важно для работы с нестационарными рядами, характер которых может резко меняться под воздействием внешних факторов [3].

В данной работе предлагается контекстно-зависимый метод адаптивной настройки параметров авторегрессионных моделей для задач интерполяции. Его основное отличие заключается в том, что алгоритм выбора параметров (p, d, q) использует два ключевых контекстных фактора, специфичных для области пропуска: длительность пропуска в данных и количественную оценку интенсивности внешних возмущений в соответствующий период. В качестве демонстрации эффективности предложенного метода и валидации подхода он применен к задаче восстановления реальных геомагнитных данных, характеризующихся выраженной

нестационарностью и подверженных влиянию солнечной активности, что моделируется через соответствующий геомагнитный индекс. Целью исследования является демонстрация принципиальной возможности и преимущества включения локальной контекстной информации в процесс параметризации моделей для интерполяции временных рядов [4], что открывает путь к созданию более надежных и точных аналитических инструментов для восстановления данных в различных предметных областях.

2. Состояние вопроса. Развитие методов автоматического анализа временных рядов непосредственно связано с потребностью в эффективных инструментах для обработки неполных данных, возникающих в реальных системах мониторинга. Исторически проблема восстановления пропусков решалась либо простыми статистическими методами (линейная интерполяция, заполнение средним), либо с помощью скользящих средних. Однако эти подходы не учитывали автокорреляционную структуру данных, что приводило к значительным искажениям при работе с нестационарными рядами, характерными для геофизических, экономических и инженерных измерений.

Переломным моментом стало широкое внедрение в практику методологии Бокса–Дженкинса и класса авторегрессионных интегрированных моделей скользящего среднего (ARIMA), которые за счет оператора дифференцирования позволяют работать с нестационарностью, а компоненты авторегрессии и скользящего среднего эффективно идентифицируют внутреннюю динамику процесса [5]. Это создало теоретическую основу для использования данных моделей не только для прогнозирования, но и для интерполяции пропусков.

Следующим логическим шагом стала автоматизация трудоемкого процесса идентификации и оценивания параметров ARIMA-моделей. Алгоритмы, подобные реализованному в пакете forecast для R [1], стали промышленным стандартом, демонстрируя высокую эффективность в условиях, когда временной ряд может быть описан единой структурой на всем периоде наблюдений. Эти алгоритмы, основанные на комбинации статистических тестов и информационных критериев, минимизируют необходимость экспертного вмешательства. Однако их фундаментальное ограничение проистекает из самой цели их создания – они оптимизированы для глобального описания ряда в целях прогнозирования. При этом задача локального восстановления пропуска, особенно в условиях изменяющейся внешней среды, предъявляет иные требования к модели.

Параметры, обеспечивающие минимальную ошибку на всей исторической выборке, могут оказаться субоптимальными для аппроксимации поведения системы в конкретный, потенциально аномальный, интервал времени, отмеченный пропуском данных.

Расширение ARIMA до моделей с экзогенными переменными (ARIMAX) стало попыткой учесть влияние внешних факторов [6]. Этот подход показал свою эффективность в ситуациях, когда существует четко измеряемый внешний драйвер, коррелирующий с основным рядом. Тем не менее, его применимость для восстановления пропусков ограничена необходимостью наличия полных данных по экзогенным переменным за весь период интерполяции, что на практике часто невыполнимо. Более того, ARIMAX не решает проблему адаптивного выбора структурных параметров (p, d, q) – модель получает внешний вход, но ее архитектура остается фиксированной и подобранной глобально.

В последние годы исследовательский интерес сместился в сторону машинного обучения и гибридных моделей. Работы, подобные [4], демонстрируют потенциал комбинирования линейных авторегрессионных моделей с нелинейными аппроксиматорами, такими как искусственные нейронные сети, для учета сложных паттернов. Однако возрастающая сложность таких моделей часто требует больших объемов данных для обучения, может вести к переобучению и снижает интерпретируемость результатов. Кроме того, вопросы адаптивного выбора гиперпараметров и архитектуры гибридных моделей в зависимости от локального контекста пропуска остаются открытыми.

Параллельно в отдельных работах (например, [7]) отмечается, что характеристики самого пропуска (механизм возникновения, длина, расположение) являются критически важной мета-информацией. Эмпирически установлено, что точность большинства методов интерполяции снижается с увеличением длины пропуска. Тем не менее, эта зависимость редко формализуется в виде явного алгоритмического правила для перенастройки модели. Существующие решения, как классические, так и современные, действуют в парадигме, где алгоритм либо не принимает во внимание эту информацию, либо пассивно наблюдает ухудшение качества, но не меняет свою стратегию для компенсации данного эффекта.

Таким образом, обзор современных подходов выявляет существующую методологическую проблему. Недостаточно разработанными остаются методы, которые активно использовали бы контекстную информацию о пропуске и внешних условиях для динамической адаптации структурных параметров базовой модели, а не

только ее коэффициентов. Требуется переход от принципа «одна модель для всего ряда» к концепции «адаптивная модель для конкретного пропуска».

3. Описание предлагаемого метода. В основе предлагаемого подхода лежит принцип контекстно-зависимой адаптации [8], согласно которому параметры авторегрессионной модели, используемой для интерполяции пропуска, должны определяться не глобальными свойствами всего временного ряда, а локальными условиями, характерными для данного конкретного пропуска.

Данный принцип реализуется в виде формального алгоритма, который интегрирует два ключевых фактора контекста: количественную характеристику самого пропуска и меру интенсивности внешних возмущений в соответствующий период. Это позволяет перейти от модели с фиксированной структурой к адаптивной процедуре, где архитектура модели является функцией от контекста:

$$M = f(L, I), \tag{1}$$

где M – выбираемая модель ARIMA(p, d, q), L – длительность пропуска, I – индекс внешней активности.

Первый контекстный фактор, длительность пропуска L (измеряемая в количестве отсчетов временного ряда, где для минутных данных 1 отсчет = 1 минута), непосредственно влияет на сложность задачи восстановления. Эмпирически установлено, что с увеличением L авторегрессионная составляющая теряет предсказательную силу из-за ослабления корреляционной связи с граничными известными значениями. Для учета этого эффекта в предложенном подходе вводится эвристическое правило, связывающее максимальный допустимый порядок авторегрессии $p_{\max}$ с длиной пропуска:

$$p_{\max} = \max(1, \lfloor k / L \rfloor), \tag{2}$$

где k – эмпирический коэффициент, характеризующий типичный временной масштаб (лаг) значимой автокорреляции ряда. Он оценивается, например, как наибольший лаг, на котором модуль выборочной автокорреляционной функции (АКФ) ряда в окрестности пропуска превышает заданный порог значимости. Конструкция формулы обеспечивает, что $p_{\max}$ является целым положительным числом: обозначение $\lfloor k / L \rfloor$ означает взятие целой части от деления, а функция max(1, ...) гарантирует, что порядок авторегрессии будет не менее 1, даже если $k < L$ (т.е. при очень длинных пропусках или слабой

автокорреляции ряда). Это предотвращает попытку построения модели с $p = 0$, которая была бы бессмысленна в данном контексте.

Данное ограничение предотвращает переобучение модели на малом объеме релевантных данных, доступных для обучения в локальном окне.

Второй и наиболее значимый с точки зрения новизны фактор – это индекс внешней активности I, характеризующий интенсивность возмущений в системе в период возникновения пропуска. В качестве индекса внешней активности I может использоваться любой количественный дескриптор состояния внешней среды, коррелирующий с динамикой исследуемого процесса (например, индекс солнечной активности для геофизических данных, объем торгов для финансовых рядов, показатель нагрузки для инженерных систем).

Предполагается, что уровень I коррелирует с характером нестационарности ряда. В периоды высокой внешней активности (высокое I) процесс может демонстрировать поведение, близкое к белому шуму с резкими скачками, что требует увеличения веса компоненты скользящего среднего (MA) в модели.

Для формализации указанной зависимости предлагается использовать нормализованный индекс

$$I_{norm} = \frac{I - I_{\min}}{I_{\max} - I_{\min}}, \tag{3}$$

на основе которого вычисляется эвристический весовой коэффициент ω для смещения баланса между AR и MA компонентами в пространстве параметров.

Коэффициент ω определяется логистической функцией:

$$\omega(I_{norm}) = \frac{1}{1 + \exp(-\alpha(I_{norm} - \beta))}, \tag{4}$$

где параметры α и β калибруются на валидационной выборке. В логистической функции параметр $\beta \in (0, 1)$ задает пороговое значение I_{norm}, при котором $\omega = 0.5$, а параметр $\alpha > 0$ определяет скорость перехода функции от 0 к 1. Их конкретные оптимальные значения находятся методом поиска по сетке в процессе калибровки метода для конкретного типа данных.

Высокое значение ω (близкое к 1) указывает на предпочтительность моделей с повышенным порядком q.

Используя входные параметры L и I, алгоритм формирует ограниченное пространство допустимых моделей C. Это пространство представляет собой подмножество всевозможных троек (p, d, q), отобранное по правилам:

$$p \in [1, p_{\max}], \tag{5}$$

где $d \in \{0, 1, 2\}$ (определяется предварительным тестом Дики-Фуллера на стационарность остатков в локальном окне [9, 10]), q выбирается из диапазона, смещенного в зависимости от ω(I_{norm}):

$$q \in [q_{\min}, q_{\max}], \tag{6}$$

где $q_{\min} = \max(1, \lfloor \omega Q \rfloor)$, $q_{\max}$ – константа, задающая верхнюю границу. Здесь ω – весовой коэффициент из (4), Q – положительная масштабирующая константа, определяющая чувствительность порядка скользящего среднего q к изменению коэффициента ω. Конкретное значение Q, как и $q_{\max}$, выбирается на этапе калибровки метода и зависит от типичного диапазона порядков q, адекватных для моделирования данного типа рядов.

Таким образом, контекстные факторы не предписывают жестко единственную модель, а сужают область поиска до наиболее правдоподобных с точки зрения текущих условий допустимых моделей.

Ключевым этапом подхода является процедура кросс-валидации на смежных данных для выбора окончательной модели из множества C. Для этого в окрестностях пропуска, на известных данных, искусственно создается валидационный пропуск той же длины L. Для каждой модели из C производится ее обучение на усеченном ряду (с искусственным пропуском) и последующая интерполяция этого пропуска. Качество интерполяции оценивается на известных значениях, которые были искусственно скрыты. В качестве целевой функции оптимизации $Q(M_i)$ используется взвешенная комбинация среднеквадратичной ошибки (RMSE) и информационного критерия Акаике (AIC) [11], которая позволяет учитывать как точность аппроксимации, так и сложность модели, предотвращая излишнее усложнение:

$$Q(M_i) = \gamma RMSE_{norm} + (1 - \gamma) AIC_{norm}, \tag{7}$$

где $RMSE_{norm}$ и AIC_{norm} – нормализованные значения метрик, γ – весовой коэффициент.

Модель M_{opt}, доставляющая минимум функции Q, выбирается для финального восстановления целевого пропуска. Оптимальная модель M_{opt} (p_{opt}, d_{opt}, q_{opt}), доставляющая минимум функции Q, выбирается для финального восстановления целевого пропуска.

Представленные эмпирические правила (2) и (4) не являются теоретически выведенными, а представляют собой содержательную эвристику, направленную на решение двух ключевых практических проблем при восстановлении пропусков в реальных нестационарных рядах. Правило (2), связывающее максимальный порядок авторегрессии p_{max} с длиной пропуска L, ограничивает структурную сложность модели при дефиците релевантных данных для обучения, предотвращая переобучение. Правило (4), определяющее весовой коэффициент ω через индекс внешней активности I, адаптирует баланс между авторегрессионной и скользящей средней компонентами модели к изменяющемуся уровню стохастических внешних возмущений.

Калибровка гиперпараметров алгоритма – коэффициентов α и β логистической функции (4) и весового коэффициента γ целевой функции (7) – выполнялась на выделенной валидационной выборке исторических данных. Для оптимизации использовался метод полного перебора (Grid Search) [8] по предопределенным сеткам значений: параметр α варьировался в диапазоне от 5 до 20 с шагом 1, параметр β – в диапазоне от 0.3 до 0.7 с шагом 0.05. Данные диапазоны были определены эмпирически на основе предварительных экспериментов и охватывают область, в которой логистическая функция (4) демонстрирует плавный, но отчетливый переход выходного значения ω от состояния, близкого к 0 (для AR-компоненты), к состоянию, близкому к 1 (для MA-компоненты), на всем диапазоне нормализованного индекса I_{norm} от 0 до 1. Критерием оптимизации служило максимальное значение среднего коэффициента детерминации (R^2), достигнутое на множестве валидационных пропусков. Весовой коэффициент γ, управляющий балансом между ошибкой (RMSE) и сложностью модели (AIC) в выражении (7), был подобран аналогичным образом; его оптимальное значение составило 0.7. Этот процесс обеспечил объективный и воспроизводимый выбор гиперпараметров [12], адаптированных к специфике анализируемых геомагнитных рядов.

Подобный подход к адаптивному управлению сложностью модели согласуется с общей практикой в прикладном анализе данных и машинном обучении, где строгий теоретический вывод часто

дополняется или заменяется эмпирически обоснованными правилами для работы с конкретными данными. Обоснованность и адекватность данных эвристик проверяются в рамках процедуры кросс-валидации, описанной выше (например, выбор модели на основе минимизации функции $Q(M_i)$ в (7)), где они непосредственно влияют на формирование пространства множества допустимых моделей C и, как следствие, на итоговое качество восстановления. Правило (2) основывается на статистическом принципе, согласно которому для устойчивой оценки параметров авторегрессии порядка p требуется объем выборки, существенно превышающий p. Отношение k/L является упрощенной оценкой количества доступных независимых наблюдений на один оцениваемый параметр, что напрямую связано с проблемой переобучения. Ключевые параметры правил – коэффициент k в (2), а также α и β в (4) – не задаются априори, а калибруются на отдельной валидационной выборке для конкретного типа данных, что делает метод принципиально адаптируемым к различным предметным областям и условиям наблюдений.

После выбора оптимальной модели $M_{\text{opt}}(p_{\text{opt}}, d_{\text{opt}}, q_{\text{opt}})$ производится заключительный этап – непосредственная интерполяция исходного пропуска длины L. Модель обучается на всем доступном сегменте данных, окружающем пропуск, который включает N точек до и после него. Обученная модель ARIMA затем используется для получения оценок пропущенных значений.

Рассмотрим модель ARIMA(p, d, q), заданную разностным уравнением:

$$(1-B)^d y_t = c + \sum_{i=1}^{p} \varphi_i (1-B)^d y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \, \varepsilon_{t-j}, \qquad (8)$$

или в эквивалентной операторной форме:

$$\nabla^d y_t = c + \sum_{i=1}^{p} \varphi_i \nabla^d y_{t-i} \, \varepsilon_t + \sum_{j=1}^{q} \theta_j \, \varepsilon_{t-j}, \qquad (9)$$

где ∇^d – оператор дифференцирования порядка d, y_t – значение ряда в момент t, ε_t – белый шум, φ_i (i = 1,…, p) и θ_j (j = 1,…, q) – параметры авторегрессии и скользящего среднего, c – константа модели.

Для (9) восстановление значений внутри пропуска осуществляется последовательным вычислением условного

математического ожидания. Этот процесс эквивалентен односторонней фильтрации, при которой каждое последующее восстанавливаемое значение вычисляется с учетом как ранее предсказанных значений внутри пропуска, так и известных исторических данных и оценок ошибок.

Алгоритмическая реализация формальных правил, заданных выражениями (1)-(7), требует практической адаптации к особенностям реальных данных, которые часто демонстрируют сложные нестационарные паттерны, не укладывающиеся в строгие теоретические предположения. Для устойчивой работы алгоритма на этапе предварительной обработки входной временной ряд подвергается процедуре мягкого сглаживания и очистки от выбросов. Это необходимо для стабилизации оценок локальной статистики, таких как автокорреляционная функция и дисперсия, которые критически важны для формул (2) и (3). В частности, значение эмпирического коэффициента k в уравнении (2) корректируется с учетом локальной волатильности ряда в окрестности пропуска. На спокойных участках с низкой дисперсией можно допустить использование более высокого порядка p, так как даже слабые корреляционные связи могут быть информативными. Напротив, в турбулентных сегментах, где шумовая компонента велика, соотношение (2) ужесточается, чтобы избежать попыток моделирования случайных флуктуаций, что предотвращает переобучение. Эта динамическая корректировка делает адаптацию к первому контекстному фактору (L) не механической, а учитывающей качество доступной для обучения информации.

Калибровка параметров логистической функции (4), связывающей нормализованный индекс активности I_{norm} с весовым коэффициентом ω, представляет собой отдельную задачу оптимизации [13]. Для ее решения создается специальная валидационная выборка, состоящая из множества исторических пропусков с известными истинными значениями. На этой выборке методом поиска по сетке определяются значения α и β, которые максимизируют общее качество восстановления. При этом качество понимается не только как минимизация среднеквадратичной ошибки, но и как способность модели сохранять важные структурные особенности сигнала, такие как экстремумы и точки перегиба, которые часто теряются при неудачном выборе параметров. Этот итеративный процесс настройки превращает уравнение (4) из абстрактной зависимости в конкретный инструмент, настроенный на специфику предметной области, будь то геомагнитные данные или другой тип сигналов. В результате, для спокойных периодов [14] ($I_{\text{norm}} \rightarrow 0$)

алгоритм склоняется к выбору моделей с более выраженной авторегрессионной компонентой, что хорошо согласуется с теорией о возможности более длинной памяти у стабильных процессов. В периоды высокой возмущенности ($I_{\text{norm}} \to 1$) возрастающая ω смещает предпочтения алгоритма в сторону моделей с высоким порядком q, что позволяет более гибко адаптироваться к резким, похожим на шум изменениям, характерным для таких условий.

4. Программная реализация и архитектура предлагаемого решения. Разработанный метод был реализован в виде специализированного программного модуля на языке Python, выбранного ввиду его распространенности в научных вычислениях и наличию развитой экосистемы библиотек для анализа данных.

Ядро модуля построено по объектно-ориентированной архитектуре [15], центральным элементом которой является класс ContextAwareImputer. Такой дизайн инкапсулирует всю логику контекстно-зависимой настройки, предоставляя пользователю простой программный интерфейс для восстановления пропусков в виде метода impute(data, gap_indices, context_I).

Для обеспечения вычислительной эффективности и воспроизводимости результатов в реализации активно задействованы библиотеки NumPy, pandas и statsmodels [16 – 18]. Последняя предоставляет надежную реализацию оценки и прогнозирования для моделей ARIMA, которая была интегрирована и расширена в рамках предлагаемого подхода. В частности, стандартный класс ARIMA из statsmodels был обернут в процедуру, которая динамически меняет его параметры order в соответствии с алгоритмом, описанным в разделе 3.

Критически важным аспектом реализации является управление вычислительной сложностью. Полный перебор всех возможных троек (p, d, q) даже в суженном пространстве моделей C, определенном выражениями (5) и (6), может стать ресурсоемкой операцией при обработке длинных рядов или большого количества пропусков. Для оптимизации этого процесса в алгоритм внедрен механизм кеширования.

Результаты предварительного теста Дики–Фуллера на стационарность, а также вычисленные автокорреляционные функции для стандартных сегментов данных сохраняются и повторно используются, что позволяет избежать дублирующих вычислений. Кроме того, процедура кросс-валидации для оценки возможных моделей, заданная выражением (7), была распараллелена. Используя возможности библиотеки joblib, оценка качества каждой модели из множества C производится асинхронно на доступных ядрах процессора,

что приводит к почти линейному ускорению вычислений на многоядерных системах. Этот подход делает метод практичным для работы с данными большого объема.

Входными данными для модуля являются одномерный массив числовых значений (временной ряд), массив индексов или меток времени, определяющих начало и конец каждого пропуска, и соответствующий массив значений контекстного индекса I для каждого пропуска.

Программный конвейер обработки начинается с этапа валидации и предварительной очистки входных данных. На этом этапе проверяется согласованность длин массивов, осуществляется мягкое сглаживание ряда для подавления высокочастотного шума, не несущего информации для структурного анализа, и выполняется первичное обнаружение и обработка выбросов. Затем для каждого пропуска определяется его локальное окно – сегмент данных, окружающий пропуск, размер которого пропорционален длине пропуска L, но ограничен сверху для сохранения локальности анализа. Именно в пределах этого окна вычисляются все необходимые статистики:

– оценка автокорреляционной функции для расчета коэффициента k в соответствии с выражением (2),

– локальная дисперсия для корректировки коэффициента k.

Кроме того проводится тест Дики–Фуллера для принятия решения о порядке дифференцирования d.

Архитектурно модуль разделен на несколько логических компонентов (рисунок 1). Компонент ContextAnalyzer отвечает за прием исходных данных, их предобработку и вычисление параметров L и I_{norm} для каждого пропуска. ParameterSpaceConstructor реализует правила, заданные уравнениями (2), (4), (5) и (6), преобразуя пару (L, I_{norm}) в конкретное множество моделей C. ModelSelector выполняет процедуру кросс-валидации, описанную в разделе 3, включая создание искусственного валидационного пропуска, обучение моделей и вычисление целевой функции качества $Q(M_i)$. Наконец, компонент ImputationEngine, получив от ModelSelector оптимальные параметры (p_{opt}, d_{opt}, q_{opt}), осуществляет финальное обучение модели ARIMA на полном локальном окне и выполняет последовательную интерполяцию пропущенных значений, используя механизм условного прогноза, основанный на выражении (9).

Такая модульная архитектура не только улучшает читаемость и поддерживаемость кода, но и облегчает его возможное расширение, например, для интеграции других типов прогнозных моделей или дополнительных контекстных факторов.

Для обеспечения удобства использования модуль снабжен подробной документацией, сгенерированной с помощью Sphinx, включающей описание API, примеры запуска и руководство по калибровке гиперпараметров алгоритма, таких как коэффициенты α, β и γ.

Расширение модуля для обработки различных типов временных рядов потребовало создания универсального механизма адаптации базовых гиперпараметров. Исходные значения коэффициентов α и β в логистической функции (4), а также веса γ в целевой функции (7) были получены в ходе калибровки на тестовых данных.

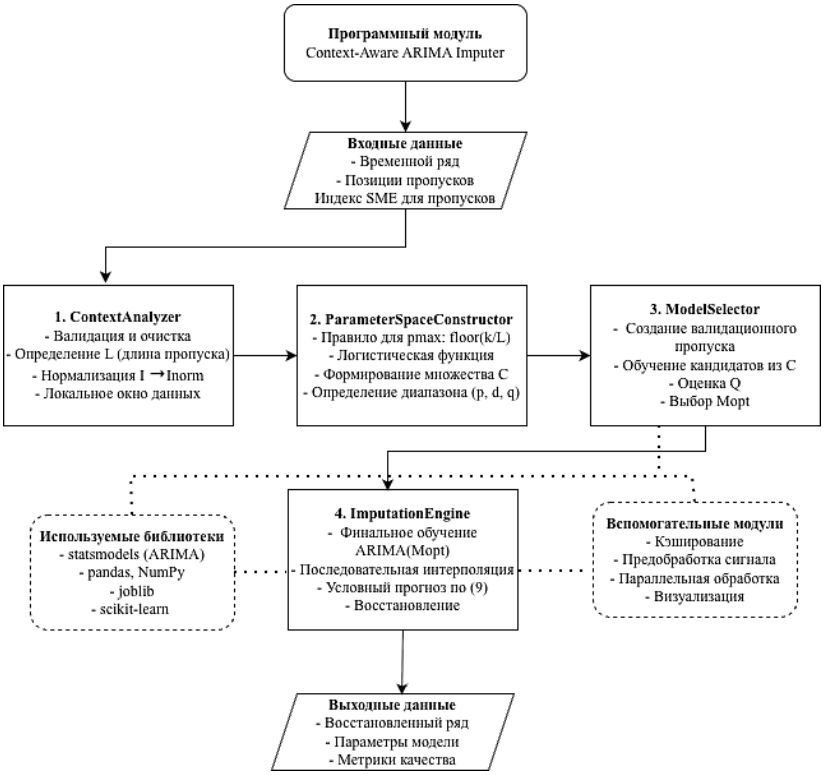


Рис. 1. Взаимодействие компонентов в методе адаптивной параметризации. Сплошные стрелки обозначают основной поток данных и управления между ядром алгоритма. Пунктирные стрелки обозначают зависимость компонентов от внешних библиотек

Однако для применения метода к рядам из других предметных областей необходима возможность их тонкой настройки. В архитектуру модуля был включен специализированный класс HyperparameterOptimizer, который через конфигурационный файл позволяет пользователю задавать границы для автоматизированного поиска оптимальных значений этих параметров для своего набора данных. Процедура поиска основана на минимизации средней ошибки восстановления на выделенной калибровочной выборке с использованием метода случайного поиска по сетке, что обеспечивает баланс между эффективностью и вычислительными затратами. Этот механизм формализует процесс адаптации ядра алгоритма к новой предметной области, сохраняя при этом его основную логику неизменной.

Одной из ключевых задач стала разработка устойчивых стратегий обработки пограничных случаев и аномальных сценариев, возникающих при работе с реальными данными. В компонентах ContextAnalyzer и ParameterSpaceConstructor реализованы многоуровневые проверки корректности входных данных и промежуточных вычислений. Например, если алгоритм обнаруживает, что локальное окно данных содержит недостаточное количество наблюдений для надежной оценки автокорреляционной функции или дисперсии, он автоматически переключается на использование глобальных оценок, вычисленных для всего ряда, или консервативных значений по умолчанию.

Аналогичным образом, в ситуациях, когда вычисленное множество моделей C оказывается пустым из-за чрезмерно жестких ограничений, правила ослабляются в итеративном режиме до тех пор, пока не будет сформировано хотя бы несколько допустимых моделей. Эти стратегии гарантируют, что алгоритм не завершит работу с фатальной ошибкой, а выдаст осмысленный, хотя и, возможно, субоптимальный результат даже в нестандартных условиях.

Обеспечение численной устойчивости и точности вычислений при оценке и прогнозировании моделей ARIMA потребовало особого внимания к настройке параметров оптимизации и обработке численных ошибок. Библиотека statsmodels, лежащая в основе вычислений, предоставляет различные алгоритмы оптимизации для подбора коэффициентов моделей (метод максимального правдоподобия). В ходе экспериментов было установлено, что выбор конкретного решателя и его гиперпараметров (допуск, максимальное число итераций) может существенно влиять на сходимость и точность оценки, особенно для моделей с высокими порядками p и q. Поэтому в компонентах

ModelSelector и ImputationEngine реализована обертка, которая отслеживает предупреждения и ошибки в процессе оптимизации. В случае сбоя или отсутствия сходимости для сложной модели, алгоритм автоматически переключается на более простой решатель или уменьшает максимальный порядок модели в множестве моделей C. Это обеспечивает надежность итоговых вычислений, предотвращая ситуации, когда весь процесс восстановления прерывается из-за неудачи на этапе обучения одной из множества проверяемых моделей.

Валидация корректности численных расчетов и логики работы всех компонентов потребовала создания комплексной системы модульных и интеграционных тестов. Тестовый набор включает синтетические данные, сгенерированные процессами ARIMA с известными параметрами, что позволяет напрямую проверять способность алгоритма восстанавливать пропуски в условиях, когда истинная модель ряда априори известна.

Модульные тесты проверяют, что каждый компонент для фиксированных входных данных выдает строго ожидаемый результат, например, что ParameterSpaceConstructor для заданных L и I_norm формирует определенное множество моделей C. Интеграционные тесты проверяют согласованность работы всего конвейера: восстановленные значения для искусственного пропуска в синтетическом ряду должны с заданной точностью совпадать с заведомо известными истинными значениями. Эта система тестов служит не только для обеспечения надежности, но и как формальная спецификация ожидаемого поведения системы в различных условиях, что критически важно для долгосрочной поддержки и развития кодовой базы.

Для обеспечения прозрачности и интерпретируемости работы сложного адаптивного алгоритма была разработана система детального логирования и генерации отчетов. Каждый основной компонент модуля записывает в структурированный лог ключевые промежуточные результаты и принятые решения: вычисленные значения L и I_norm, построенное множество моделей C, значения целевой функции $Q(M_i)$ для каждой модели, результаты проверки на сходимость и финальный выбор параметров модели. Эти данные могут быть автоматически агрегированы в текстовый или визуальный отчет, который позволяет исследователю реконструировать весь процесс обработки конкретного пропуска. Такая прозрачность превращает алгоритм из «черного ящика» в инструмент, работа которого может быть проанализирована, объяснена и, при необходимости, скорректирована, что является важным требованием для научных и инженерных приложений, где

понимание причинно-следственных связей не менее важно, чем итоговый результат.

5. Экспериментальное исследование на геомагнитных данных. Для практической проверки предложенного метода и демонстрации его работоспособности было выполнено экспериментальное исследование на реальных геомагнитных временных рядах.

Выбор геомагнитных данных в качестве тестового стенда обусловлен комплексом причин, делающих их уникальным и требовательным полигоном для алгоритмов восстановления нестационарных процессов. Во-первых, геомагнитное поле является непрерывно наблюдаемым физическим полем, чья динамика формируется под воздействием как внутренних процессов ядра Земли, так и внешних воздействий со стороны солнечного ветра и магнитосферной активности. Эта двойственная природа приводит к формированию сложного сигнала, сочетающего в себе относительно плавные суточные вариации, обусловленные вращением Земли, и резкие, импульсные возмущения (суббури, бури), связанные с солнечной активностью. Таким образом, геомагнитный ряд по своей сути является ярко выраженным нестационарным процессом со смешанным спектром, где спокойные периоды сменяются интервалами высокой турбулентности. Такое поведение представляет собой полезный, но крайне сложный случай для проверки способности алгоритма адаптироваться к фундаментальным изменениям в характере данных.

Во-вторых, для геомагнитных данных существует хорошо разработанная система количественных индексов, объективно описывающих уровень внешней возмущенности. Индекс SME (Substorm Magnetospheric Index), используемый в данном исследовании, рассчитывается по глобальной сети наземных обсерваторий и служит надежной интегральной мерой энергии, вкладываемой в магнитосферу во время суббуревых событий. Наличие такого независимого, непрерывного и количественного дескриптора внешнего контекста (фактор I в предлагаемом методе) является редким преимуществом. Во многих других областях (финансы, медицина, техника) подобные внешние факторы либо ненаблюдаемы, либо имеют качественный характер, что затрудняет их формальное использование в алгоритмах. Таким образом, геомагнитные данные предоставляют возможность проверить гипотезу о полезности интеграции внешней контекстной информации в процесс параметризации модели, поскольку здесь эта

информация доступна, измерима и имеет четкую физическую интерпретацию.

В качестве конкретного объекта исследования использовались минутные данные вариаций магнитного поля Земли, предоставляемые международной сетью обсерваторий SuperMAG [19, 20]. Эта сеть обеспечивает стандартизованную предобработку сырых измерений, включая удаление основного поля и приведение к единой системе координат, что гарантирует высокое качество и сопоставимость данных из различных источников.

Для проведения вычислительных экспериментов был использован ряд северной компоненты (DBE_NEZ), полученный на высокоширотной обсерватории Ловозеро (LOZ) Полярного геофизического института (ПГИ) [21]. Выборка была ограничена периодом 2015 года, что позволяет обеспечить сопоставимость условий наблюдений и полную доступность синхронных индексов геомагнитной активности SME. Целевым параметром, подвергаемым восстановлению, была выбрана северная компонента возмущенного магнитного поля (DBE_NEZ). Данная компонента является одним из ключевых индикаторов суббуревых процессов в высоких широтах, демонстрируя особенно сильный отклик на возмущения в магнитосферных токах. Ее динамика характеризуется значительной изменчивостью амплитуды и частоты колебаний, что создает дополнительную сложность для задач интерполяции по сравнению с более плавными компонентами.

Для моделирования условий пропусков из исходного непрерывного ряда были искусственно удалены сегменты различной длины, что позволило иметь точный эталон для объективной оценки качества интерполяции. Диапазон длин пропусков L был выбран от 5 до 120 минут с дискретным шагом (5, 10, 15, 30, 45, 60, 90, 120 минут), что соответствует длинам от 5 до 120 отсчетов при минутном разрешении данных. Этот выбор не является произвольным; он соответствует типичным временным масштабам потери данных в реальных системах мониторинга, которые могут быть вызваны кратковременными сбоями в передаче (минуты), плановым техническим обслуживанием (десятки минут) или более длительными отказами оборудования. Исследование именно таких диапазонов длин представляет наибольший практический интерес, поскольку методы простой экстраполяции или прогноза на один шаг оказываются неадекватными, а необходимость в сложной модели, учитывающей структуру ряда, становится критической.

Ключевой особенностью экспериментального плана являлась неслучайная, а систематическая привязка каждого искусственно

созданного пропуска к конкретному уровню геомагнитной активности, характеризуемому индексом SME. Вместо того чтобы случайным образом распределять пропуски по всему временному ряду, они целенаправленно размещались внутри заранее определенных временных интервалов, классифицированных по уровням SME: спокойные условия (SME < 100 нТл), слабые возмущения (100–300 нТл), умеренные бури (300–600 нТл) и сильные возмущения (SME > 600 нТл). Такой подход позволил сформировать сбалансированную тестовую выборку, равномерно покрывающую весь спектр возможных условий, в которых может работать алгоритм восстановления.

Подготовка данных к эксперименту включала несколько обязательных этапов. Исходный минутный ряд, как и любой реальный геофизический сигнал, содержал технические артефакты и выбросы, не связанные с геофизическими процессами. Для их подавления применялся мягкий фильтр на основе алгоритма Савицкого–Голея, который эффективно удаляет высокочастотный шум, минимально искажая форму основного сигнала. Важно отметить, что эта предобработка применялась только к данным, используемым для обучения и тестирования моделей; исходные значения сохранялись в качестве эталона для расчета финальных метрик ошибки. Это позволяет оценить, насколько хорошо восстановленные значения соответствуют реальным наблюдаемым данным, а не их сглаженной версии. После очистки проводилась проверка ряда на стационарность с помощью расширенного теста Дики–Фуллера (ADF), которая подтвердила наличие единичного корня, то есть нестационарность ряда. Это обосновывает необходимость использования именно класса ARIMA-моделей.

Планирование эксперимента дало возможность проверить не только общую эффективность метода в терминах средней ошибки, но и его ключевую концептуальную гипотезу – способность адаптивно и осмысленно менять параметры модели в зависимости от контекста, определяемого парой (L, SME). Для этого в ходе работы алгоритма для каждого тестового пропуска протоколировался финальный выбор оптимальной модели $M_{\text{opt}}(p_{\text{opt}}, d_{\text{opt}}, q_{\text{opt}})$. Последующий статистический анализ этих выборов позволил выявить устойчивые паттерны: например, тенденцию к выбору моделей с более высоким порядком q (скользящее среднее) в периоды высокой активности SME, что соответствует физическому ожиданию о возрастающей роли стохастических, похожих на шум возмущений. Или тенденцию к уменьшению допустимого порядка p (авторегрессии) с ростом длины пропуска L, что свидетельствует о корректной работе эвристического

правила, ограничивающего сложность модели при дефиците релевантной информации.

Для каждого тестового случая работа предложенного контекстно-зависимого метода сравнивалась с двумя базовыми подходами, представляющими разные философии восстановления данных. Первый базовый метод – классическая кусочно-линейная интерполяция – представляет собой простейший детерминированный подход, полностью игнорирующий как автокорреляционную структуру ряда, так и внешний контекст. Он служит нижним примером, демонстрирующим минимальный ожидаемый уровень качества, который должен быть превзойден любым более сложным методом.

Второй метод для сравнения – интерполяция с помощью модели ARIMA, параметры которой (p, d, q) были однократно подобраны глобальным алгоритмом auto.arima по всему доступному временному ряду (за исключением самого пропуска). Этот метод представляет собой современный стандарт де-факто в автоматическом анализе временных рядов. Однако, будучи примененным к задаче интерполяции, он воплощает философию «одна модель для всех случаев»: структура модели, выбранная на основе глобальных свойств всего ряда, используется для восстановления любого пропуска независимо от его локальных особенностей и условий внешней среды. Сравнение с этим методом позволяет количественно оценить ценность, которую добавляет контекстно-зависимая адаптация, предлагаемая в данной работе.

Качество восстановления оценивалось по двум взаимодополняющим метрикам: коэффициенту детерминации (R^2) и среднеквадратичной ошибке (RMSE), вычисленным путем сравнения восстановленных значений с исходными, необработанными («сырыми») данными внутри каждого искусственного пропуска. Использование R^2 в качестве основной метрики позволяет оценить, какая доля дисперсии исходного сигнала объясняется восстановленными значениями, в то время как RMSE дает понимание типичной величины отклонения в абсолютных единицах (нТл). Качественная иллюстрация работы метода в сложных условиях представлена на рисунке 2, который демонстрирует восстановление 60-минутного пропуска в спокойной обстановке (SME $\approx$ 80 нТл) и позволяет визуально сравнить подходы.

Количественные результаты эксперимента позволяют сделать следующие выводы об эффективности предлагаемого метода в зависимости от контекста. В спокойных и слабовозмущенных условиях (SME = 50–200 нТл) метод демонстрирует стабильно высокое качество

восстановления, с коэффициентом детерминации R^2 = 0.71–0.85 для всего диапазона длин пропусков от 5 до 120 мин. Это подтверждает его надежность в режимах, где динамика ряда относительно предсказуема.

В условиях умеренных возмущений (SME ≈ 400 нТл) точность заметно снижается, достигая значений R^2 = 0.23–0.49, что отражает возрастающую стохастичность сигнала. Наиболее сложным случаем для восстановления оказались периоды сильных и экстремальных бурь (SME ≥ 800 нТл), где качество падает до R^2 = 0.15–0.41, что является фундаментальным ограничением, связанным с приближением исходного ряда к шуму при высокой внешней возмущенности. Интересно отметить, что для самых длинных пропусков (90–120 мин) в этих экстремальных условиях метод иногда показывает сопоставимую или даже чуть более высокую точность, чем для коротких пропусков, что может указывать на сглаживающий эффект модели на длинных интервалах при очень хаотичном сигнале. Эти результаты наглядно иллюстрируют как сильные стороны метода – его адаптивность и надежность в широком диапазоне условий, так и объективные границы его применимости, определяемые природой исходных данных.

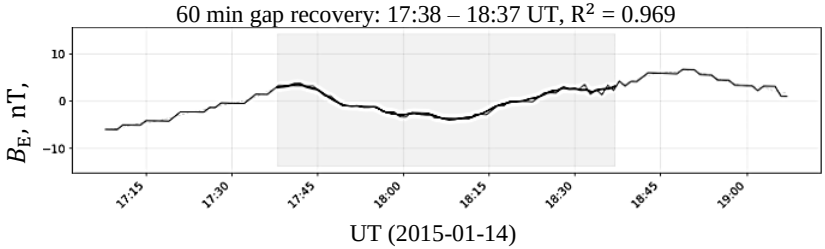


Рис. 2. Пример восстановления 60-минутного пропуска dbe_nez при SME = 80 нТл

Для каждого тестового случая работа предложенного контекстно-зависимого метода сравнивалась с двумя базовыми подходами, реализующими разные принципы восстановления данных. Первый, кусочно-линейная интерполяция, служил простейшим детерминированным ориентиром и, как и ожидалось, показывал резкое снижение точности с ростом длины пропуска (например, с $R^2 \approx 0.91$ для 5-минутного до $R^2 < 0.20$ для 30-минутного интервала). Второй, более совершенный подход – интерполяция с помощью модели ARIMA, параметры которой (p, d, q) были однократно подобраны глобальным алгоритмом `auto.arima` по всему ряду, – воплощал принцип «одна модель для всех случаев». Однако в условиях умеренных геомагнитных

возмущений (SME ≈ 400 нТл) предложенный адаптивный метод для коротких пропусков (5–15 мин) более чем вдвое превосходил эту глобальную ARIMA по точности (R^2 = 0.79–0.85 против R^2 = 0.39–0.46), а в экстремальных условиях (SME ≥ 800 нТл) сохранял преимущество в 30–60 процентных пунктов по R^2. Даже в спокойной обстановке (SME 50–200 нТл) адаптивный метод поддерживал стабильно высокое качество (R^2 = 0.83–0.94) на длинных интервалах (30–60 мин), где линейная интерполяция уже не работала. Эти результаты количественно подтверждают, что переход от универсальной к адаптивной, контекстно-зависимой параметризации модели обеспечивает существенный и устойчивый выигрыш в точности восстановления данных, особенно в условиях внешних возмущений.

6. Обсуждение результатов. Результаты экспериментального исследования демонстрируют, что предложенный контекстно-зависимый метод обеспечивает новый подход к задаче интерполяции пропусков в нестационарных временных рядах по сравнению с классическими решениями. Полученные данные не просто подтверждают его работоспособность, но и позволяют глубоко проанализировать механизмы его работы и границы эффективности. Ключевым практическим выводом является доказанная способность алгоритма поддерживать высокое качество восстановления ($R^2 > 0.7$) в широком диапазоне длин пропусков (5–120 мин) при условии спокойной или слабовозмущенной геомагнитной обстановки. Это указывает на то, что метод успешно решает выявленную проблему: он эффективно использует локальную автокорреляционную структуру данных, адаптивно ограничивая сложность модели в зависимости от доступного для анализа объема информации. Стабильность результатов на длинных пропусках особенно важна, так как именно в таких сценариях традиционные методы, основанные на экстраполяции, терпят неудачу.

Вместе с тем, экспериментальные данные ясно показывают фундаментальную зависимость качества восстановления от уровня внешней возмущенности, характеризуемого индексом SME. Резкое снижение коэффициента R^2 при SME > 300 нТл является не недостатком алгоритма, а отражением объективного физического ограничения. В периоды высокой геомагнитной активности динамика параметра DBE_NEZ становится крайне турбулентной, приближаясь к поведению окрашенного шума с резкими, плохо прогнозируемыми скачками.

Полученные результаты также позволяют четко очертить область наиболее эффективного применения метода. Его сильные стороны максимально раскрываются при работе с нестационарными рядами,

которые, однако, демонстрируют относительно устойчивую автокорреляционную структуру в пределах локального окна анализа. Это характерно для данных мониторинга физических процессов в штатных режимах их работы. В этих условиях метод обеспечивает точное, структуро-сохраняющее восстановление пропусков различной длины. С другой стороны, в периоды экстремальных возмущений, когда ряд теряет выраженную автокорреляцию, метод, как и любой другой, основанный на линейном прогнозе, достигает своего теоретического предела точности. Однако важно отметить, что даже в этих условиях он не «ломается», а выдает консервативный результат, часто превосходящий по точности простую линейную интерполяцию.

Стоит подчеркнуть, что предложенный метод носит универсальный характер и не ограничивается геомагнитными данными. Данный конкретный случай был выбран для экспериментальной проверки в первую очередь в силу его доступности для авторов и наличия в нем всех ключевых атрибутов, необходимых для валидации метода: выраженной нестационарности и наличия формализованного внешнего индекса активности (SME). Успешное применение алгоритма в этих сложных условиях служит убедительным доказательством его работоспособности. Модульная архитектура алгоритма и его ключевые принципы (учет длительности пропуска L и внешнего контекста I) заведомо готовы к адаптации для работы с временными рядами из других предметных областей (например, финансовой аналитики, мониторинга технологических процессов), где могут быть определены аналогичные контекстные факторы.

Представляется возможным несколько перспективных направлений для дальнейшего развития метода. Во-первых, интеграция более сложных моделей, учитывающих нелинейные зависимости, например, в рамках гибридного подхода, где контекстно-зависимый механизм выбирал бы не только параметры, но и класс модели (например, между линейной ARIMA и нелинейной моделью на основе деревьев). Во-вторых, использованием не одного, а нескольких контекстных индексов, которые могли бы более тонко описывать состояние системы. Наконец, разработанная модульная архитектура программной реализации позволяет относительно легко адаптировать ядро алгоритма для работы с данными из других предметных областей, таких как финансовая аналитика или мониторинг промышленного оборудования, где также существуют проблемы пропусков и доступны внешние индикаторы состояния рынка или технологического процесса. В этом смысле предлагаемое решение закладывает основу для создания универсального адаптивного инструментария восстановления данных,

способного учитывать специфику контекста в самых разных приложениях.

Авторы благодарят рецензентов за внимательное прочтение работы и конструктивную критику, которая позволила существенно улучшить изложение и методологическую строгость представленных результатов.

Литература

1. Januschowski T., Gasthaus J., Wang Y., Salinas D., Flunkert V., Bohlke-Schneider M., Callot L. Criteria for classifying forecasting methods // International Journal of Forecasting. 2020. vol. 36. no. 1. pp. 167–177. DOI: 10.1016/j.ijforecast.2019.05.008.
2. pmdarima: Arima estimators for python. Online Code Repos. Available at: http://www.alkaline-ml.com/pmdarima. (accessed 26.02.2026).
3. Hamilton J.D. Time Series Analysis. Princeton: Princeton University Press, 2020. 816 p. DOI: 10.2307/j.ctv14jx6sm.
4. Lama A., Ray S., Biswas T., Narsimhaiah L., Raghav Y.S., Kapoor P., Singh K.N., Mishra P., Gurung B. Python code for modeling ARIMA-LSTM architecture with random forest algorithm // Software Impacts. 2024. vol. 20. DOI: 10.1016/j.simpa.2024.100650.
5. Jiang Y., Ning K., Pan Z., Shen X., Ni J., Yu W., Schneider A., Chen H., Nevmyvaka Y., Song D. Multi-modal time series analysis: A tutorial and survey. Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2025. vol. 2. pp. 6043–6053. DOI: 10.1145/3711896.3736567.
6. Kowal D. Dynamic Regression Models for Time-Ordered Functional Data // Bayesian Analysis. 2021. vol. 16(2). pp. 459–487. DOI: 10.1214/20-BA1213.
7. Bokde N.D., Yaseen Z.M., Andersen G.B. ForecastTB – An R Package as a Test-Bench for Time Series Forecasting – Application of Wind Speed and Solar Radiation Modeling. Energies 2020. vol. 13. no. 10. DOI: 10.3390/en13102578.
8. Vorobev A.V., Vorobeva G.R. An approach to dynamic visualization of heterogeneous geospatial vector images // Computer Optics. 2024. vol. 48(1). pp. 123–138. DOI: 10.18287/2412-6179-CO1279.
9. Maitra S., Politis D.N. Prepivoted Augmented Dickey-Fuller Test with Bootstrap-Assisted Lag Length Selection. Stats. 2024. vol. 7(4). pp. 1226–1243. DOI: 10.3390/stats7040072.
10. Akaike H. A new look at the statistical model identification. IEEE Transactions on Automatic Control. 1974. vol. 19. no. 6. pp. 716–723. DOI: 10.1109/TAC.1974.1100705.
11. Hill C., Du L., Johnson M., McCullough B.D. Comparing programming languages for data analytics: Accuracy of estimation in Python and R. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2024. vol. 14(3). DOI: 10.1002/widm.1531.
12. Kataoka R. Extreme geomagnetic activities: a statistical study. Earth Planets Space. 2020. vol. 72(1). DOI: 10.1186/s40623-020-01261-8.
13. Boroyev R.N., Vasiliev M.S. The auroral activity during the main phase of magnetic storms. Advances in Space Research. 2023. vol. 71(1). pp. 1137–1145. DOI: 10.1016/j.asr.2022.10.034.
14. Newell P.T., Gjerloev J.W. Evaluation of SuperMAG auroral electrojet indices as indicators of substorms and auroral power. Journal of Geophysical Research: Space Physics. 2011. vol. 116. DOI: 10.1029/2011JA016779.

15. Chu X., Ma D., Bortnik J., Tobiska W.K., Cruz A., Bouwer S.D., et al. Relativistic electron model in the outer radiation belt using a neural network approach. Space Weather. 2021. vol. 19. pp. 1–18. DOI: 10.1029/2021SW002808.
16. Gupta P., Bagchi A. Data Manipulation with Pandas. Essentials of Python for Artificial Intelligence and Machine Learning. Synthesis Lectures on Engineering, Science, and Technology. 2024. pp. 197-235. DOI: 10.1007/978-3-031-43725-0_6.
17. Sundaram J., Gowri K., Devaraju S., Gokuldev S., Jayaprakash S., Anandaram H., Manivasagan C., Thenmozhi M. An Exploration of Python Libraries in Machine Learning Models for Data Science. 2023. pp. 1–31. DOI: 10.4018/978-1-6684-8696-2.ch001.
18. Faraway J.J. Linear Models with Python. Boca Raton, FL: Chapman and Hall/CRC, 2021. 308 p.
19. Gjerloev J.W. A Global Ground-Based Magnetometer Initiative. Eos, Transactions American Geophysical Union. 2009. vol. 90(27). pp. 230–231. DOI: 10.1029/2009EO270002.
20. Gjerloev J.W. The SuperMAG data processing technique. Journal of Geophysical Research: Space Physics. 2012. vol. 117. DOI: 10.1029/2012JA017683.
21. PGI Geophysical data. January, February, March 2013. Murmansk, Apatity: PGI KSC RAS, 2013.

Воробьев Андрей Владимирович — д-р техн. наук, профессор, заведующий кафедрой, кафедра информатики, Уфимский университет науки и технологий. Область научных интересов: геоинформационные технологии, цифровая обработка сигналов. Число научных публикаций — 200. geomagnet@list.ru; улица Карла Маркса, 12, 450000, Уфа, Россия; р.т.: +7(917)345-2299.

Воробьева Гульнара Равилевна — д-р техн. наук, профессор кафедры, кафедра вычислительной математики и кибернетики, Уфимский университет науки и технологий. Область научных интересов: геоинформационные и веб-технологии, системы хранения и обработки информации. Число научных публикаций — 163. gulnara.vorobeva@gmail.com; улица Карла Маркса, 12, 450000, Уфа, Россия; р.т.: +7(917)417-4111.

Поддержка исследований. Работа выполнена при поддержке Российского научного фонда (проект № 21-77-30010-П).

DOI 10.15622/ia.25.2.4

A. VOROBEV, G. VOROBEVA

A CONTEXT-DEPENDENT METHOD FOR ADAPTIVE TUNING OF PARAMETERS OF AUTOREGRESSIVE MODELS FOR NON-STATIONARY TIME SERIES

Vorobev A., Vorobeva G. **A Context-Dependent Method for Adaptive Tuning of Parameters of Autoregressive Models for Non-Stationary Time Series.**

Abstract. A method for context-sensitive tuning of autoregressive model parameters for gap reconstruction in nonstationary time series is proposed. A key feature of the method is the adaptive selection of ARIMA (p, d, q) model parameters based on two context factors: the gap duration and the level of external disturbances during the corresponding period. Unlike standard automatic model selection approaches focused on global optimization for forecasting, the developed algorithm narrows the model search space and selects the optimal configuration using local cross-validation, allowing for consideration of specific conditions in the gap region. The method is implemented as a Python software module with a modular architecture that ensures computational efficiency through caching and parallel computing. The effectiveness of the method was tested experimentally on real geomagnetic data (the DBE_NEZ component of the Lovozero Observatory). The results demonstrate that under calm and weakly disturbed geomagnetic conditions (SME index = 50–200 nT), the method provides high reconstruction accuracy (R^2 = 0.71–0.85) for gaps ranging from 5 to 120 minutes in length. However, accuracy is shown to decrease consistently with increasing disturbance level, reflecting a fundamental limitation associated with the increasing stochasticity of the original signal. The proposed approach ensures interpretability and adaptability, opening up prospects for the development of data reconstruction tools in various application areas.

Keywords: autoregressive models, gap reconstruction, nonstationary time series, geomagnetic data.

References

1. Januschowski T., Gasthaus J., Wang Y., Salinas D., Flunkert V., Bohlke-Schneider M., Callot L. Criteria for classifying forecasting methods. International Journal of Forecasting. 2020. vol. 36. no. 1. pp. 167–177. DOI: 10.1016/j.ijforecast.2019.05.008.
2. pmdarima: Arima estimators for python. Online Code Repos. Available at: http://www.alkaline-ml.com/pmdarima. (accessed 26.02.2026).
3. Hamilton J.D. Time Series Analysis. Princeton: Princeton University Press, 2020. 816 p. DOI: 10.2307/j.ctv14jx6sm.
4. Lama A., Ray S., Biswas T., Narsimhaiah L., Raghav Y.S., Kapoor P., Singh K.N., Mishra P., Gurung B. Python code for modeling ARIMA-LSTM architecture with random forest algorithm. Software Impacts. 2024. vol. 20. DOI: 10.1016/j.simpa.2024.100650.
5. Jiang Y., Ning K., Pan Z., Shen X., Ni J., Yu W., Schneider A., Chen H., Nevmyvaka Y., Song D. Multi-modal time series analysis: A tutorial and survey. Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2025. vol. 2. pp. 6043–6053. DOI: 10.1145/3711896.3736567.
6. Kowal D. Dynamic Regression Models for Time-Ordered Functional Data. Bayesian Analysis. 2021. vol. 16(2). pp. 459–487. DOI: 10.1214/20-BA1213.
7. Bokde N.D., Yaseen Z.M., Andersen G.B. ForecastTB – An R Package as a Test-Bench for Time Series Forecasting – Application of Wind Speed and Solar Radiation Modeling. Energies 2020. vol. 13. no. 10. DOI: 10.3390/en13102578.

8. Vorobev A.V., Vorobeva G.R. An approach to dynamic visualization of heterogeneous geospatial vector images. Computer Optics. 2024. vol. 48(1). pp. 123–138. DOI: 10.18287/2412-6179-CO1279.
9. Maitra S., Politis D.N. Prepivoted Augmented Dickey-Fuller Test with Bootstrap-Assisted Lag Length Selection. Stats. 2024. vol. 7(4). pp. 1226–1243. DOI: 10.3390/stats7040072.
10. Akaike H. A new look at the statistical model identification. IEEE Transactions on Automatic Control. 1974. vol. 19. no. 6. pp. 716–723. DOI: 10.1109/TAC.1974.1100705.
11. Hill C., Du L., Johnson M., McCullough B.D. Comparing programming languages for data analytics: Accuracy of estimation in Python and R. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2024. vol. 14(3). DOI: 10.1002/widm.1531.
12. Kataoka R. Extreme geomagnetic activities: a statistical study. Earth Planets Space. 2020. vol. 72(1). DOI: 10.1186/s40623-020-01261-8.
13. Boroyev R.N., Vasiliev M.S. The auroral activity during the main phase of magnetic storms. Advances in Space Research. 2023. vol. 71(1). pp. 1137–1145. DOI: 10.1016/j.asr.2022.10.034.
14. Newell P.T., Gjerloev J.W. Evaluation of SuperMAG auroral electrojet indices as indicators of substorms and auroral power. Journal of Geophysical Research: Space Physics. 2011. vol. 116. DOI: 10.1029/2011JA016779.
15. Chu X., Ma D., Bortnik J., Tobiska W.K., Cruz A., Bouwer S.D., et al. Relativistic electron model in the outer radiation belt using a neural network approach. Space Weather. 2021. vol. 19. pp. 1–18. DOI: 10.1029/2021SW002808.
16. Gupta P., Bagchi A. Data Manipulation with Pandas. Essentials of Python for Artificial Intelligence and Machine Learning. Synthesis Lectures on Engineering, Science, and Technology. 2024. pp. 197-235. DOI: 10.1007/978-3-031-43725-0_6.
17. Sundaram J., Gowri K., Devaraju S., Gokuldev S., Jayaprakash S., Anandaram H., Manivasagan C., Thenmozhi M. An Exploration of Python Libraries in Machine Learning Models for Data Science. 2023. pp. 1–31. DOI: 10.4018/978-1-6684-8696-2.ch001.
18. Faraway J.J. Linear Models with Python. Boca Raton, FL: Chapman and Hall/CRC, 2021. 308 p.
19. Gjerloev J.W. A Global Ground-Based Magnetometer Initiative. Eos, Transactions American Geophysical Union. 2009. vol. 90(27). pp. 230–231. DOI: 10.1029/2009EO270002.
20. Gjerloev J.W. The SuperMAG data processing technique. Journal of Geophysical Research: Space Physics. 2012. vol. 117. DOI: 10.1029/2012JA017683.
21. PGI Geophysical data. January, February, March 2013. Murmansk, Apatity: PGI KSC RAS, 2013.

Vorobev Andrei — Ph.D., Dr.Sci., Professor, Head of the department, Informatics Department, Ufa University of Science and Technology. Research interests: geoinformation technologies, digital signal processing. The number of publications — 200. geomagnet@list.ru; 12, Karl Marx St., 450000, Ufa, Russia; office phone: +7(917)345-2299.

Vorobeva Gulnara — Ph.D., Dr.Sci., Professor of the department, Computational Mathematics and Cybernetics Department, Ufa University of Science and Technology. Research interests: geoinformation and web technologies, systems of information storing and processing. The number of publications — 163. gulnara.vorobeva@gmail.com; 12, Karl Marx St., 450000, Ufa, Russia; office phone: +7(917)417-4111.

Acknowledgements. This work was funded by the Russian Science Foundation (project No. 21-77-30010-P).

DOI 10.15622/ia.25.2.5

N.V. HUNG, P.D. HUYNH, M.V. TUNG, N.V. VU, N.P. DAT

CLVM: A HYBRID DEEP LEARNING FRAMEWORK FOR CONTACTLESS VIRTUAL MOUSE CONTROL

Nguyen Viet Hung, Phi Dinh Huynh, Ma Van Tung, Nguyen Van Vu, Nguyen Phu Dat. **CLVM: A Hybrid Deep Learning Framework for Contactless Virtual Mouse Control.**

Abstract. In the era of rapid digital transformation and the growing prevalence of artificial intelligence, enabling natural, seamless, and contactless human-computer interaction has become a critical priority across various domains. This paper presents a novel deep learning-based model for virtual mouse control using hand gestures, termed CLVM (CNN-LSTM Virtual Mouse). The proposed system introduces a hybrid architecture that integrates three powerful components: (1) MediaPipe for efficient and real-time hand landmark detection; (2) a Convolutional Neural Network (CNN) for spatial feature extraction; and (3) a Long Short-Term Memory (LSTM) network for temporal dynamics modeling, enhancing the system's ability to recognize gestures continuously and accurately over time. Unlike traditional models, CLVM is designed to maintain robust performance in real-world environments, particularly under conditions of inconsistent lighting and cluttered backgrounds. The system also provides low latency and high responsiveness and can be deployed effectively on resource-constrained devices, making it practical for widespread adoption. Experimental results demonstrate that CLVM achieves a high accuracy (99.88%) while reducing the loss to 0.38, significantly outperforming conventional gesture recognition methods. These findings highlight CLVM's potential to serve as a reliable, scalable, and efficient solution for natural gesture-based interaction. It offers a valuable step forward in the development of intelligent, user-friendly interfaces for contactless control applications.

Keywords: computer vision, contactless interface, hand landmarks, machine learning, MediaPipe, virtual mouse.

1. Introduction. A major area of research in fields such as artificial intelligence (AI), computer vision, and Human-Computer Interface (HCI) is the development of more intuitive and natural ways for people to interact with computer, as information technology and computer science continue to advance at a rapid pace [1-5]. Even though conventional peripherals such as physical keyboards and mice are still widely used, they have limitations in certain situations. Common scenarios include remote control in intelligent systems, non-contact settings such as industrial clean rooms, medical surgeries, or epidemic situations where contact must be minimized to reduce the risk of infection, and support systems for individuals with mobility impairments to enable easy interaction.

Therefore, gesture control systems – especially virtual mouse control systems that use hand gestures – have become a viable alternative to conventional peripherals [6-9]. However, there are still many significant technical obstacles to overcome before real-time hand gesture recognition systems can be implemented. These difficulties include maintaining high gesture recognition accuracy, reducing latency to meet real-time requirements

and improving the system's ability to adapt to a variety of environmental factors, such as changing lighting conditions, cluttered backgrounds, and variations of user hand sizes and shapes. Soving these issues requires not only strong algorithmic solutions but also close integration of hardware and software technologies, along with the development of deep learning architectures tailored for this purpose.

This paper proposes a new approach based on the integration of three key technological components to address the limitations of current virtual mouse control systems. The first element is the MediaPipe library [10, 11], a powerful and wildly-used computer vision framework that enables efficient and accurate real-time hand landmark detection Convolutional Neural Network (CNN) [12], which focuses on processing and interpreting spatial features of image data. The third element is a Long-Short-Term Memory network (LSTM) [13], which enables the system to recognize dynamic patterns in hand gestures by learning and modeling temporal sequences.

By integrating these three technologies into a single architecture, the system is able to provide users with a seamless and efficient interactive experience, achieving high gesture recognition accuracy and maintaining uninterrupted operation even in challenging real-world scenarios. To enhace practical applicability domains such as online education, remote healthcare, factory automation, and user support, the primary objective of the research is to create a virtual mouse control model that can be readily deployed on standard devices such as personal computers (PCs) or laptops with integrated webcams.

Experimental results show that the proposed model outperforms baseline methods in a number of aspects, including high-precision classification, inference time, and adaptability to a variety of usage scenarios.

The objectives of this research can be summarized in three main directions. First, we seek to place CLVM within the broader landscape of state-of-the-art (SOTA) gesture recognition systems. While many existing approaches rely on rule-based designs or use either CNNs or LSTMs in isolation, our framework combines the two to model both spatial and temporal aspects of hand gestures. This integration is intended to improve robustness and maintain continuity during real-time interaction. Second, beyond technical accuracy, we consider the practical aspects of user experience. Although this work primarily emphasizes performance metrics (accuracy, latency, stability), preliminary informal trials indicate that the defined gestures are intuitive and easy to perform, and we identify structured user evaluations as a critical next step. Third, we acknowledge that the current system supports only four fundamental gestures (Move, Scroll, Pause, Start). This deliberate choice allows us to

rigorously validate feasibility under real-world conditions. Simultaneously, it highlights the path for future work, where the gesture vocabulary will be expanded to include more complex actions such as clicking, dragging, and zooming, toward building a fully functional replacement for traditional mouse devices. Together, these goals establish CLVM not merely as a proof-of-concept, but as a robust and extensible framework that addresses both algorithmic and usability challenges in natural human-computer interaction.

The proposed system is illustrated in Figure 1. It shows the overall architecture of the contactless virtual mouse system.

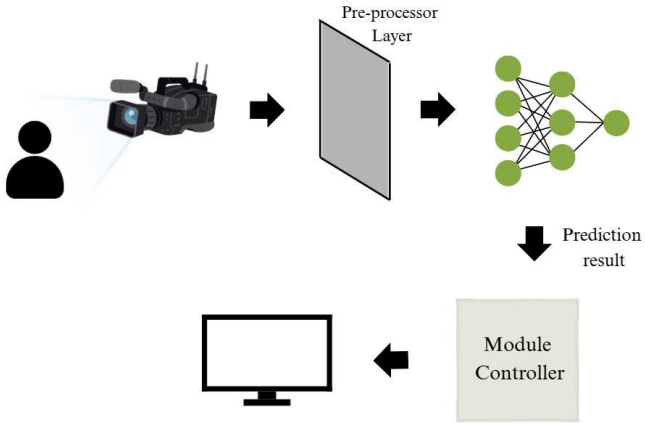


Fig. 1. System overview diagram

Initially, a camera device (such as a head-mounted camera or smart glasses) captures the user's hand movements or gestures. The captured data are then passed through a pre-processing layer, which is responsible for normalizing and extracting relevant features. This processed data is fed into a machine learning model or neural network that analyzes the input and produces a prediction representing the intended user action. The prediction is sent a module control, which interprets the output and converts it into corresponding mouse commands. Finally, these commands are executed on the computer system, enabling intuitive and touchless control of a virtual mouse cursor. Our target users include accessibility device users and operators in sterile or dusty workplaces, as well as public kiosk users and videoconferencing presenters. By eliminating contact with shared hardware, CLVM reduces the risk of cross-contamination and mechanical wear without sacrificing responsiveness.

Based on the proposed system architecture, this paper introduces the following key innovations:

– We propose CLVM (CNN-LSTM Virtual Mouse) – a novel optimization framework for gesture-based virtual mouse control – by integrating convolutional and recurrent neural networks. This hybrid design enables efficient extraction of both spatial features (via CNN) and temporal dynamics (via LSTM), overcoming the limitations of traditional single-model approaches.

– The proposed three-stage pipeline integrates: (1) MediaPipe to detect hand landmarks in real time, (2) a CNN to learn spatial gesture representations, and (3) an LSTM to model sequential patterns for robust, continuous recognition.

– Experimental results show that CLVM outperforms baseline models (CNN-only, LSTM-only, and conventional rule-based approaches) in accuracy and robustness. Specifically, CLVM achieves an average gesture recognition accuracy of 99.88% under diverse conditions, surpassing prior work that typically reports accuracies ranging from 99.08% to 99.80% in controlled environments. The system maintains performance even in low light and cluttered backgrounds, demonstrating strong generalization.

The remainder of this paper is organized as follows. Section 2 reviews related work to establish the context and highlight the gap our study addresses. In Section 3, we introduce our proposed method, CLVM (CNN-LSTM Virtual Mouse), including the overall system architecture and its key components. Specifically, we describe the use of 1D convolutional layers for local feature extraction, LSTM networks for temporal sequence modeling, and the final integration and output stage. Section 4 presents the performance evaluation of our method based on relevant benchmarks. Finally, Section 5 concludes the paper and outlines potential directions for future work.

2. Related work. This section provides a summary of previous and ongoing research projects that are directly relevant to our study. Within the field of HCI, these studies focus on developing touchless interaction solutions and virtual mouse control systems with the goal of improving the user experience.

One study [14] proposed a computer vision-based virtual mouse system that controls the cursor through hand movements instead of a conventional physical mouse. This system, implemented in Python, uses the OpenCV library to recognize and track the user's hand motions in real time via a camera [15-17]. Its primary features include double-click, right-click, and left-click functionalities, providing the user a variety of control options.

Another study [18] proposed a system that extends the virtual mouse control approach by combining voice commands with hand gestures to provide more convenient and natural interaction. The system uses the pyttsx3 package

for voice synthesis, along with MediaPipe and OpenCV for hand gesture recognition [11, 19, 20], allowing users to execute commands precisely via voice.

A further study [21] developed a real-time camera-based mouse control system that integrates voice assistance with hand gestures. The aim of this system is to enhance the efficiency, comfort, and naturalness of human-computer interaction. Its optimized algorithms, along with natural language control via an integrated voice assistant, significantly enhace the user experience.

Notably, recent studies on computer mouse control have focused on using artificial intelligence (AI) focused, leading to smoother and more natural mouse movements. Most earlier studies [9, 14] frequently employ the MediaPipe library for hand gesture recognition often under specific conditions.

Simultaneously, studies in related areas, such as virtual reality [22, 23] or frame estimation [24, 25], are also rapidly evolving with the aim of helping to accelerate processing speed and reduce system resource consumption. This trend indicates growing attention to spatial development.

Additionally, significant advancement has been made in this area through Virtual Whiteboard (VW) technology, which was introduced in [23, 26, 27]. VW technology translates hand motions into digital tasks such as writing, sketching, and form editing using computer vision and machine learning algorithms. The system efficiently meets the demands of note-taking and illustration in real time by supporting a variety of drawing tools, enabling fast PDF export, and automatically converting handwriting into typed text.

Another study [28] focuses on creating an interactive presentation. This system uses OpenCV and CvZone to recognize hand gestures and enables users to annotate content and control slides directly via a webcam [28-30]. In lectures or seminars, this approach enhances audience interaction and presenter flexibility by eliminating the need for keyboards or remote controls.

In a study [31], a virtual mouse system was proposed that uses a webcam and deep learning models to recognize hand movements and translate them into mouse commands without physical contact with the goal of increasing accessibility. By reducing contact, this method not only reduces the risk of infection but also assists people with disabilities who can use their hands and arms.

Further broadening the application range, a hands-free contactless virtual mouse system [32] combines facial gestures (e.g., mouth and eyes) to improve accuracy and expressiveness for users with limited hand function. The system fully supports keystrokes, mouse clicks, and page scrolling, thereby enhancing user independence. It employs advanced image processing techniques to mitigate errors caused by varying lighting.

To improve information security and prevent unauthorized access, research [33] combines virtual mouse technology with cutting-edge security strategies such as biometric authentication and encryption. This system guarantees the security of sensitive user data while also increasing accessibility in human-machine interaction.

The Enhanced Hand Tracking (EHT) model [34] utilizes the YOLOV8 architecture to improve the accuracy of real-time hand tracking and gesture recognition. This system outperforms MediaPipe, adapts to individual users' gesture patterns, supports multi-user interaction, and is tailored for virtual reality environments. Thus, EHT promises to provide a smooth, natural, and highly customized interaction experience in digital settings.

In addition to visual-only cursor control, previous works have explored bimodal interfaces that combine computer vision with automatic speech recognition to separate spatial pointing from command semantics. For example, Karpov et al. integrate head pointing for 2D cursor movements with voice commands and evaluate the interface using the ISO 9241-9 methodology, demonstrating the practicality of contactless pointing with speech [35]. Previous studies compare two speech and gesture systems (ICANDO, MOWGLI) and report Fitts' law experiments for multimodal pointing, highlighting the fusion time and user performance trade-offs [36]. More broadly, surveys on multimodal HCI promote combining modalities to reduce errors and improve robustness in real-world settings [37]. In our work, CLVM is positioned as a vision-centric module that leverages MediaPipe hand landmarks for real-time control and can be extended with voice commands for mode switching or click/drag actions in noisy or hands-busy situations [38].

3. Proposed Method – CLVM. This section presents the CLVM model, a contactless virtual mouse system that combines a Convolutional Neural Networks (CNN) for hand gesture recognition and Long Short-Term Memory (LSTM) network for temporal gesture tracking. The system enables real-time cursor control using hand movements, offering high accuracy and adaptability across various lighting conditions. CLVM provides a natural, intuitive interaction method, especially beneficial for touchless interfaces and accessibility applications.

3.1. System Model. Developing a hand gesture recognition system is a complex task that requires in-depth knowledge, system design abilities, and sophisticated data processing skills. To ensure precise, reliable, and consistent recognition of pre-defined key hand gestures, such as Move, Scroll, Pause, and Start, the system is meticulously designed, undergoes several implementation stages, and is carefully optimized.

Every phase of the development process, including preliminary data collection, data preprocessing, model design, training, parameter tuning, and final performance evaluation step, is crucial and interdependent. Creating a sufficiently rich dataset that reflects realistic variations in gestures, users, and environmental conditions is necessary for the data collection step. The suitable algorithm must be chosen and optimized during the model design and training phase to achieve the best accuracy and guarantee real-time performance. Finally, the evaluation process shows the system's efficacy and dependability in real-world scenarios, where a variety of intricate and unpredictable factors exist, in addition to testing the system's accuracy on the training data.

The proposed hand gesture recognition system achieves high effectiveness, accuracy, and robustness in real-world human-machine interaction through tight integration and coordinated operation across all core components. From the initial stages of data collection and preprocessing to feature extraction, gesture classification, and real-time deployment, each phase has been meticulously crafted, thoroughly refined, and experimentally assessed. This well-structured and unified framework enables the system to deliver consistent performance not only in laboratory settings but also in unpredictable, real-world environments, serving as a dependable and intuitive interface for natural human-computer interaction.

Each step of the system is illustrated in Figure 2, starting with raw video data collection, a fundamental step to the entire model-building and training process. The goal of this step is to collect a sufficiently large, rich, and diverse dataset to train a machine learning model that can reliably and accurately recognize common hand gestures.

Figure 3 visualizes the gestures – *Move*, *Scroll*, *Pause*, and *Start* – as a distribution of feature data points in a two-dimensional feature space. Each point represents a particular recording session from the processed video input, and the color indicates the associated behavior label. The inherent complexity of natural behavior recognition results in some overlap between gesture classes in the scatter plot underscoring the significance of deep learning models in identifying and extracting latent features.

To improve the model's generalizability, video data must be collected under a variety of conditions, including different lighting levels (natural, low, artificial) and camera angles (frontal, oblique, top-down).

This approach enables the model to better adapt to circumstances outside the training environment, reducing its dependency on specific conditions and thereby increasing the system's practicality and reliability in real-world scenarios. Furthermore, the model's capacity to process new data outside

of the training environment is enhanced when the training data is rich and representative of real-world variations.

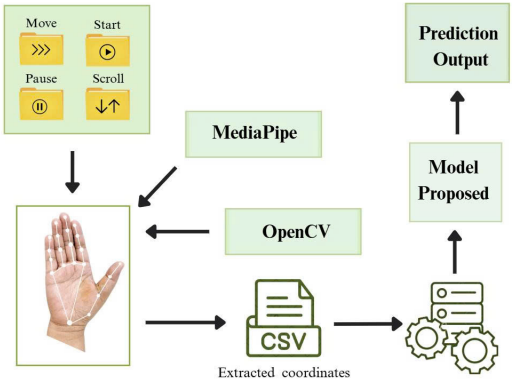


Fig. 2. Model construction diagram

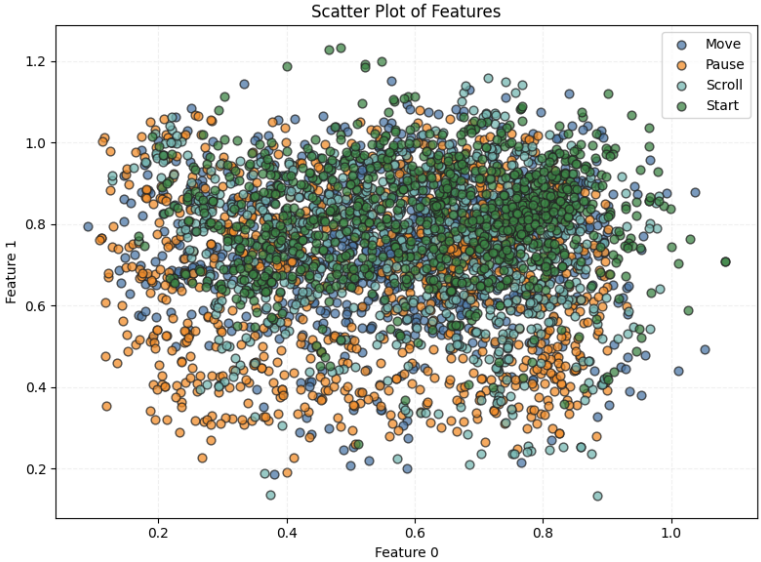
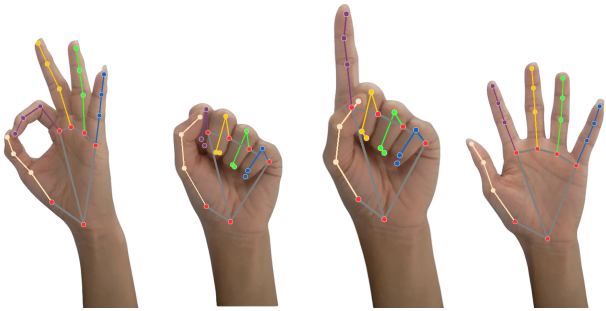


Fig. 3. Data Partition of Model

For this study, we constructed a dedicated dataset tailored to the requirements of the contactless virtual mouse application. The dataset comprises 521 short video clips, each corresponding to a single, well-defined gesture instance. Data were collected from four volunteers (three men and one woman), all right-handed. The participants exhibited natural variations in hand size, shape, and skin tone, thereby contributing to the diversity and robustness of the dataset. Each participant recorded at least 30 clips for each of the four gesture categories – Move, Scroll, Pause, and Start – resulting in a nearly uniform class distribution. The dataset was partitioned as follows: 365 clips for training, 104 for validation, and 52 for testing. The test subset remained strictly unseen during the training and validation phases. During evaluation, the gesture clips were processed using the same preprocessing pipeline adopted for training. A prediction was considered correct only when the inferred gesture label matched the corresponding ground-truth annotation. Evaluation is conducted at the window/sequence level (window length $T=10$ frames, stride $s=1$). Thus, the 52 test clips a larger number of test windows ($N_{\text{test}} \gg 52$). A prediction is counted as correct only when the inferred label matches the ground truth. To ensure consistency and reproducibility across all configurations, we use a fixed hold-out split (without k-fold cross-validation). Illustrations of the four gesture classes are provided in Figure 4.



a) Starting b) Pausing c) Moving d) Scrolling

Fig. 4. Gesture-based action descriptions (Start/Pause/Move/Scroll)

Move is a dynamic gesture. The cursor direction is determined from the frame-by-frame displacement vector of the index finger tip, which is smoothed using an Exponential Moving Average (EMA) and removing noise with a *dead-zone*; the mapping from the smoothed displacement to the cursor step follows (1)–(3). Left/Right/Up/Down correspond to the signs of the

displacement components, while diagonal directions follow the angle of the vector; **Start/Pause** is a static gesture to open/close the control loop, while **Scroll** is dynamic but only uses the vertical component.

$$\tilde{\mathbf{p}}_t = \alpha\,\mathbf{p}_t + (1-\alpha)\,\tilde{\mathbf{p}}_{t-1}, \qquad \alpha \in (0,1); \tag{1}$$

$$\Delta\tilde{\mathbf{p}}_t = \tilde{\mathbf{p}}_t - \tilde{\mathbf{p}}_{t-1}, \qquad r_t = \left\|\Delta\tilde{\mathbf{p}}_t\right\|_2; \tag{2}$$

$$\mathbf{c}_t = \gamma \max(0, r_t - \delta)\,\frac{\Delta\tilde{\mathbf{p}}_t}{\max(r_t, \varepsilon)}, \tag{3}$$

where $\mathbf{p}_t \in [0,1]^2$ are the normalized coordinates of index finger tip in frame t; $\tilde{\mathbf{p}}_t$ is the smoothed version; δ is *dead-zone*; γ is the amplification factor; $\varepsilon > 0$ prevents division by 0.

Recordings were performed under a range of conditions to test the model's ability to adapt beyond controlled laboratory settings. Lighting conditions included natural daylight (approximately 500 lux), typical indoor fluorescent lamps (around 300 lux), and a dim LED setup (close to 100 lux). Backgrounds alternated between a plain uniform wall and a cluttered office scene containing furniture and other objects. To add further variation, the camera was positioned in three different ways: directly in front of the participant (0°), at an oblique angle of about 45°, and from an overhead top-down view (90°).

All clips were recorded at a resolution of 640×480 pixels with a frame rate of 30 frames per second. The duration of each video ranged from two to five seconds. During preprocessing, horizontal flipping was applied to create both left- and right-hand variations, and all frames were converted from OpenCV's BGR format to the standard RGB format.

Unlike public gesture datasets such as SHREC, EgoHands, or NVGesture, which are oriented toward sign language or virtual reality scenarios, this dataset was specifically to reflect mouse-control actions. By tailoring the dataset to the target application, the training and evaluation of CLVM remained directly relevant to its use in real-world human-computer interaction.

Once a rich, representative, and extensive collection of videos covering the range of hand gestures in various real-world scenarios is gathered, the system proceeds to the preprocessing stage, where data are prepared for model training. This step is crucial to ensure that the input data are reliable, consistent, and suitable for the deep learning algorithms. This preprocessing phase consists of three primary tasks, each crucial for improving and normalizing the data:

– **Resizing:** data inconsistencies may arise from videos that were gathered from various sources and recorded with different resolutions and frame rates. To address this, all video frames are normalized to a fixed size of 640×480 pixels, ensuring a uniform input format. Consistent image sizes minimize computational expenses and prevent training errors, which not only guarantees uniformity, but also enhances the processing efficiency of deep learning models.

– **Image flipping:** the training dataset can be enhanced using this widely-used data augmentation method without requiring the collection of new data. The system can generate new variations of the same gesture by flipping video frames horizontally. For example, it can change the "Move" gesture from the right hand to the left. This technique helps models become less dependent on the specific hand used (right/left) and improves their generalization capability and performance on diverse real-world data.

– **Color format conversion:** ensuring a consistent color format for image data is a seemingly simple but crucial detail. While many deep learning frameworks, such as TensorFlow or PyTorch, require input data in the RGB (Red, Green, Blue) format, image processing libraries, such as OpenCV, frequently use the BGR (Blue, Green, Red) color format. To ensure compatibility and prevent errors in image data processing and analysis, all images are converted from BGR to RGB.

After preprocessing, when the input data conform to the required size, color format, and multidimensional calculation standards, the system moves on to the next stage, which is to identify and track the hand points in each video frame. For this task, we employ the **MediaPipe Hand Landmarker**, a powerful tool for precisely identifying the 3D structure of the hand. The wrist, matching fingers, and the first finger are among the 21 key points (hand landmarks) that represent each hand. A collection of 3D coordinates (x, y, z) corresponds to each of these points:

– The location of the landmark on the image plane (along the horizontal and vertical axes of the frame) is determined by x and y.

– z represents relative depth information (approximate distance along the camera axis), which helps describe the 3D hand pose in addition to the 2D image-plane coordinates (x, y).

The close coordination between landmark detection and preprocessing forms the foundation of the training process, ensuring that the system learns robust and discriminative features for reliable recognition.

The landmark data obtained from detection and tracking serve as the primary input to the deep learning model and form the basis for evaluating the overall system efficacy. This data provides rich spatio-temporal information

that captures the hand's movements, structure, and patterns of light over time. Each layer in the deep learning model's architecture is designed to contribute to the task of accurate hand gesture recognition.

The architecture includes the following key components:

– **Proposed Layer.** This layer is specifically designed to extract features from landmark data or processed images for gesture identification. Depending on the method, this layer can be implemented as either a Recurrent Neural Network (RNN) to process the motion of landmarks over time series or a Convolutional Neural Network (CNN) to take advantage of the hand's geometric and textural features in each frame. The complexity of the gestures to be classified and the type of data determine which architecture is best.

– **Dropout Layer.** This layer helps prevent overfitting. During training, it randomly deactivates a significant proportion of neurons in the layer, forcing the network to learn more robust and generalized features rather than relying on specific neuronal co-adaptations.

– **Dense Layer.** This layer combines the extracted features from previous layers to produce the final classification. The output dense layer consists of four neurons, each corresponding to one of the predefined gesture classes: Move, Scroll, Pause, and Start. The output probability for each gesture type represents the model's degree of confidence.

In addition to designing a single architecture, the system tests, evaluates, and compares numerous neural network architectures in order to determine which one best suits the data-response characteristics and real-world application requirements. Among the primary architectural elements taken into account are:

– **CLVM.** A hybrid architecture that combines CNNs to extract spatial features with LSTMs to process temporal information. This approach is effective for dynamic gestures where both motion and shape are crucial.

– **RNN.** Basic recurrent neural networks can process sequences but struggle with long-term dependencies, limiting their ability to recognize long or complex gestures.

– **LSTM.** An improved RNN that uses a unique memory mechanism to store and access information over long sequences. LSTMs are particularly useful for complex gestures with significant motion variations or long durations.

– **GRU.** A lightweight variant of LSTM that often achieves comparable performance with lower computational complexity and faster training.

In summary, the proposed hand gesture recognition system is a tightly integrated processing pipeline that incorporates advanced deep learning techniques, hand feature extraction, and image processing. The ultimate objective is to develop a solution that can meet the demands of contemporary

human-machine interaction applications by accurately, robustly, and efficiently recognizing hand gestures in real time from video data.

3.2. Method. The contactless virtual mouse system's training phase is a crucial phase that affects the system's overall performance in terms of accuracy and stability in real-world settings. Furthermore, the efficient Algorithm 1 is crucial for maintaining the system's capacity to recognize hand gestures accurately and function flawlessly in a variety of usage scenarios.

After evaluating and testing various neural network architectures, we decided to create a solution based on a combination of CNNs and LSTMs. The strengths of both model types are fully utilized in this hybrid architecture: CNNs extract spatial features from image data or landmark sequences, while LSTMs model temporal sequential relationships, a crucial component in obtaining dynamic hand gesture recognition. Figure 5 provides a detailed description of the system's interface training procedure and method, highlighting the steps involved in processing input data, architectural modeling, and training optimization.

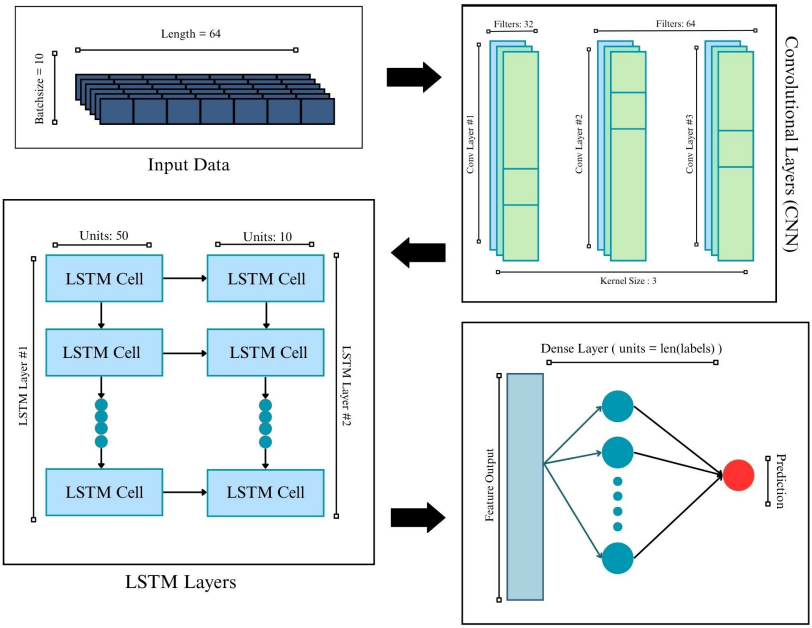


Fig. 5. System Model Method

Algorithm 1. Hand Gesture-based Mouse Control Algorithm

Input:
- Video frame F_t: Input frame from video at time t.
- Trained model M: Pre-trained model for hand gesture classification (Move/Start/Pause/Scroll).
- MediaPipe Hands mp: Hand landmark detector.
- Amplification factor α: Controls movement sensitivity.

Output: Mouse actions corresponding to detected gestures.
Initialize mouse position $(x_{prev}, y_{prev}) \leftarrow (None, None)$.
for each frame F_t from the video feed **do**
    Convert F_t to RGB image I_t.
    Detect hands H using mp on I_t.
    for each hand $h \in H$ **do**
        Extract hand landmarks L_h.
        Convert L_h to feature vector V_h.
        Predict gesture $G_h \leftarrow M(V_h)$.
        if $G_h =$ MoveMouse **then**
            Get fingertip coordinates (x, y) from L_h.
            Map (x, y) to screen coordinates and move mouse.
        else if $G_h =$ Start **then**
            Perform start virtual mouse.
        else if $G_h =$ Pause **then**
            Perform stop virtual mouse.
        else if $G_h =$ Scroll **then**
            Get index fingertip coordinates (x_{idx}, y_{idx}) from L_h.
            Map (x_{idx}, y_{idx}) to screen resolution (W, H).
            Compute scroll direction based on change in y_{idx} from previous frame.
            if $y_{prev} \neq None$ **then**
                if $y_{idx} < y_{prev}$ **then**
                    Scroll up.
                else if $y_{idx} > y_{prev}$ **then**
                    Scroll down.
                end if
            end if
            Update $y_{prev} \leftarrow y_{idx}$
        end if
    end for
    Visualize results: draw landmarks and show gesture label.
end for
return Real-time mouse control action.

To achieve robust hand gesture recognition from temporal data (such as landmark or sensor sequences), the system uses a hybrid deep learning architecture that combines LSTM units with a 1D-Dimensional Convolutional Neural Network (1D-CNN). By combining the temporal learning and memorization capabilities of LSTM with the feature extraction capabilities of CNN, the system is able to create a single end-to-end training pipeline that maximizes performance and reduces the accumulation process between intermediate stages.

1D Convolution for Local Feature Extraction. The 1D-CNN layers are designed to extract temporal feature sets from multivariate data series. Assuming the input to the network is of the form $x \in \mathbb{R}^{T \times C_{\text{in}}}$, where T is the number of time steps and C_{in} is the number of channels (or features) at each time step, the output at the time position corresponding to the k-th filter is calculated according to the formula:

$$y_t^{(k)} = \sum_{i=0}^{K-1} \sum_{j=0}^{C_{\text{in}}-1} x_{t+i,j} \cdot w_{i,j}^{(k)} + b^{(k)}, \tag{4}$$

where:

– $w_{i,j}^{(k)}$ represents the weight of the k-th kernel at position i and input channel j,

– $b^{(k)}$ is the bias term for the k-th filter,

– $x_{t+i,j}$ is the input feature value at time step $t+i$ and feature j.

Through this convolution, the network can identify subtle yet significant patterns by learning local dependencies in the input signal. The convolution's output is subsequently passed through a nonlinear activation function, usually a Rectified Linear Unit (ReLU), defined as:

$$\text{ReLU}(z) = \max(0, z). \tag{5}$$

Equation 5 introduces nonlinearity; the application of this activation function enables the network to model complex nonlinear relationships in the data, which is essential for accurate gesture classification.

To improve robustness to input variations and reduce spatial (or temporal) dimensionality, a max-pooling layer is typically applied after convulational layers. It down-samples the feature maps by selecting the maximum value in each pooling window, which provides translation invariance and helps highlight the most salient features.

The tight integration of CNN and LSTM in this framework creates a powerful and flexible deep learning model that meets the requirements for real-time hand gesture recognition with high accuracy and stable operation in large-scale human-computer interaction applications.

LSTM for Temporal Sequence Modeling. After the CNN layers extract salient local features, the LSTM layers model the temporal dependencies across successive time steps. LSTM units maintain an internal memory cell, allowing them to store and utilize information from earlier time steps in addition to processing the current input. Because of this mechanism, This mechanism enables LSTMs to learn patterns over long time horizons without being impacted by the vanishing gradient problem, a common limitation of basic RNNs on long sequences.

Given the input x_t at time step t, the hidden state h_{t-1}, and the cell state C_{t-1} from the previous time step, the LSTM updates are computed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(Forget gate)}, \tag{6}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(Input gate)}, \tag{7}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{(Candidate cell state)}, \tag{8}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad \text{(Updated cell state)}, \tag{9}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Output gate)}, \tag{10}$$

$$h_t = o_t \odot \tanh(C_t) \quad \text{(Updated hidden state)}, \tag{11}$$

where:

– $\sigma(\cdot)$ is the sigmoid activation function,

– $\tanh(\cdot)$ is the hyperbolic tangent function,

– $\odot$ denotes element-wise multiplication,

– W_f, W_i, W_C, W_o and b_f, b_i, b_C, b_o are trainable parameters for each gate.

Every gate and state in the LSTM architecture is crucial for managing information flow across time steps, enabling the learning of long-term dependencies without signal degradation.

– The **forget gate** f_t (Equation 6) determines which part of the previous memory C_{t-1} should discarded or retained. It acts as an intelligent filter that removes information no longer needed for subsequent steps, reducing memory load and preventing the accumulation of irrelevant information.

– The **input gate** i_t (Equation 7) controls what new information is added to the memory at the current time step. It works in conjunction with the candidate cell state to identify new features to learn and integrate into long-term memory.

– The **candidate cell state** $\tilde{C}_t$ (Equation 8) proposes a new memory update based on the current input and the previous hidden state. This represents the new information that the network considers incorporating into the memory.

– These three components interact in (Equation 9) to compute the new cell state C_t. Specifically, the new memory is formed by combining selected information from the old memory (via f_t) and new information added (via i_t and $\tilde{C}_t$).

– The **output gate** o_t (Equation 10) determines which part of the new cell state should be passed on as the hidden state. This serves as the summarized information that the network uses to make predictions at the current time step.

– Finally, the hidden state h_t is computed in (Equation 11) based on the output gate and the updated memory. This state is not only used to produce classification predictions at the current time step but is also passed to the next step, supporting the learning of motion sequences.

This mechanism helps the hand gesture recognition system function accurately and steadily, even with complex movement sequences, by preserving the memory required for long-term relationships and removing noise and redundant information.

Integration and Output. By integrating Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers into a unified architecture, the model is able to simultaneously capture both spatial (local) and temporal (sequential) features present in the input data. The CNN layers are responsible for extracting specific local features from individual frames, such as the shape, contour, or landmark configuration of the hand, which are key to distinguishing specific gestures. Meanwhile, the LSTM layers are responsible for modeling the dynamic evolution of these features over time, allowing the system to learn how gestures unfold as a sequence, rather than just considering static poses at each point in time. This combination significantly enhances the model's ability to distinguish between different dynamic gestures, particularly those that are similar in single frames but differ in time progression, which is crucial for the system to function correctly in a virtual mouse interface.

After processing by the CNN and LSTM layers, a dense layer with a softmax activation function receives the LSTM's final output. To translate the representation that the network has learned into probability values that correspond to each type of gesture, this layer must compute the probability

distribution over the specified gesture classes. The model is trained using the classification cross-entropy loss function, which guides the network to increase classification accuracy by measuring the degree of difference between the actual label and the predicted probability distribution. The model uses the Adam optimization algorithm, a widely-used optimization algorithm that combines advantages of AdaGrad and RMSProp, to increase training efficiency and stability and aid in faster model convergence.

Owing to its design, the system can effectively generalize to changes in user behavior, such as variations in speed, style, or consistency in gesture performance. The model is especially well-suited for interactive applications like virtual mouse control in human-machine interfaces, as it exhibits both high stability and responsiveness in real-time gesture recognition tasks.

4. Performance Evaluation. the proposed CLVM architecture was systematically evaluated using a dataset of time-series features derived from camera-captured hand gesture. These features capture the dynamics of hand movements during gesture execution, offering valuable insights into gesture patterns. Hand gestures in this study are categorized into four distinct classes: *Move*, *Pause*, *Scroll*, and *Start*. Each gesture class corresponds to a fundamental action in the virtual mouse interface, playing a critical role in enabling precise and efficient user interaction.

Accuracy, loss function, and F1-Score – three common metrics in machine learning – were used to thoroughly and impartially assess the model performance. Accuracy represents the percentage of correct predictions, Loss quantifies the discrepancy between predictions and ground truth, and the F1-Score provides a balanced measure between precision and recall, which is particularly useful for potentially imbalanced classes. To guarantee transparency and reproducibility of the results, Table 1 provides a detailed explanation of the formulas used to calculate these metrics, as well as an explanation of the relevant parameters, where:

– TP (True Positive): number of samples correctly classified into the intended gesture class.

– FN (False Negative): number of samples of that gesture class misclassified as another class.

– FP (False Positive): number of samples from other classes misclassified as that gesture class.

– TN (True Negative): number of samples from other classes correctly classified as not belonging to that gesture class.

In this part, the evaluation metrics are used to assess the models' performance on the four-class hand-gesture classification task (Move, Pause, Scroll, and Start) to ensure that the outcomes are thorough and impartial.

Table 1. Evaluation parameters

	Positive Prediction	**Negative Prediction**
Positive Action	TP	FN
Negative Action	FP	TN

Accuracy is one of the most fundamental and widely used metrics among them. We report standard classification metrics – Accuracy, Precision, Recall, and F_1-Score – computed per class and summarized by macro/micro averages [39-41]. The formulas are presented in Equations (12)-(15).

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}, \tag{12}$$

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{13}$$

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{14}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{15}$$

Table 2 summarizes the performance evaluation results, comparing the proposed model against several reference architectures. The compared models include CNN, LSTM, GRU, and RNN architectures. All these architectures are widely used in deep learning, particularly for time-series processing tasks such as hand gesture recognition.

To ensure the stability and objectivity of the outcomes, we deploy and assess each model through four separate test runs. For ease of visual comparison, accuracy and F1-Score are reported as percentages (%), while the loss is reported as the (unitless) cross-entropy value. In addition to increasing the reliability of the results obtained, performing multiple runs and averaging helps to reduce the impact of random factors in the training and evaluation process.

The results in Table 2 highlight the superior performance of the proposed CLVM architecture compared to individual LSTM, GRU, and RNN models.

The experimental results demonstrate that the proposed CLVM model outperforms the reference models. CLVM achieved an accuracy of 99.88% and

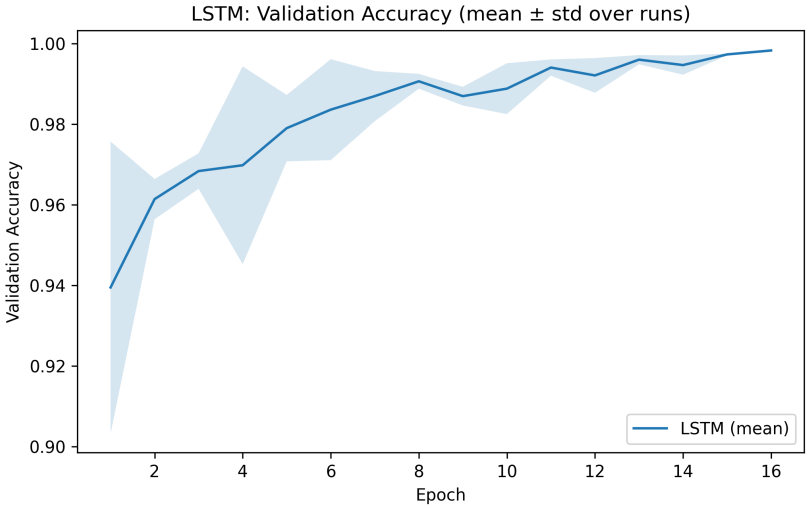
an F1-Score of 99.88%, demonstrating an excellent balance between precision and recall. Furthermore, the low loss of the model of only 0.38 indicated strong learning capacity and training stability.

Table 2. System evaluation results

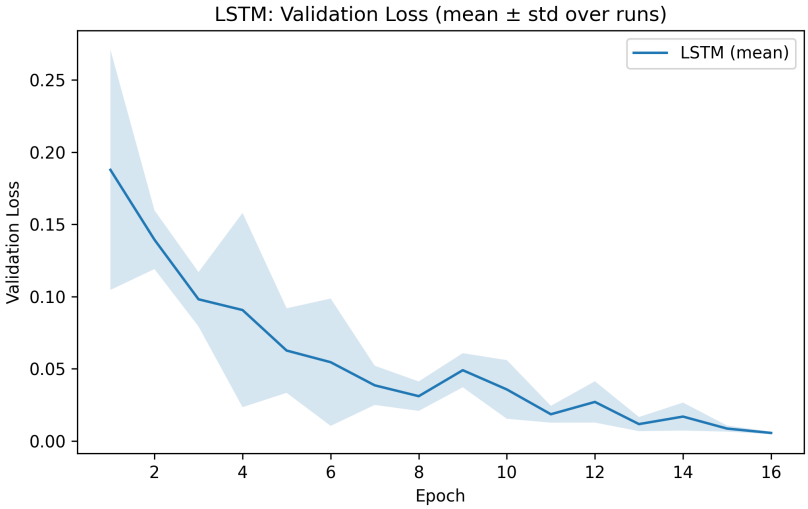
Model	Round	Accuracy (%)	Loss	F1
LSTM	Round 1	99.88	0.39	99.89
	Round 2	99.85	0.46	99.86
	Round 3	99.69	1.00	99.69
	Round 4	99.78	0.67	99.80
Average		**99.80**	**0.63**	**99.81**
GRU	Round 1	99.82	0.57	99.83
	Round 2	99.76	0.87	99.78
	Round 3	99.79	0.78	99.81
	Round 4	99.75	0.67	99.76
Average		**99.78**	**0.72**	**99.79**
RNN	Round 1	99.25	3.12	99.28
	Round 2	99.23	2.99	99.26
	Round 3	98.82	4.15	98.91
	Round 4	99.03	3.51	99.09
Average		**99.08**	**3.44**	**99.13**
CLVM	Round 1	99.93	0.26	99.93
	Round 2	99.88	0.32	99.87
	Round 3	99.82	0.57	99.83
	Round 4	99.90	0.39	99.90
Average		**99.88**	**0.38**	**99.88**

These results clearly demonstrate that the proposed CLVM system outperforms not only the individual models (LSTM, GRU, RNN) but also achieves top scores across all three evaluation metrics. This underscores the superiority of the hybrid CLVM architecture for processing and classifying gesture sequences in real-time virtual mouse interfaces.

Future work will involve evaluating CLVM on richer datasets with a wider variety of real-world conditions to further validate its generalization capability for complex contactless interaction scenarios. The performance difference between the CLVM hybrid architecture that we propose and conventional sequential models is evident from the obtained results. The training and testing curves for each model are depicted in Figures 6 – 9.

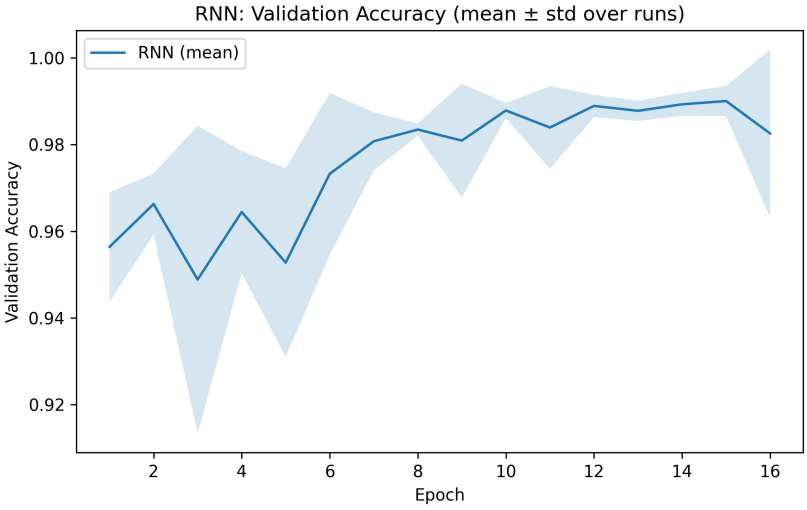


a)

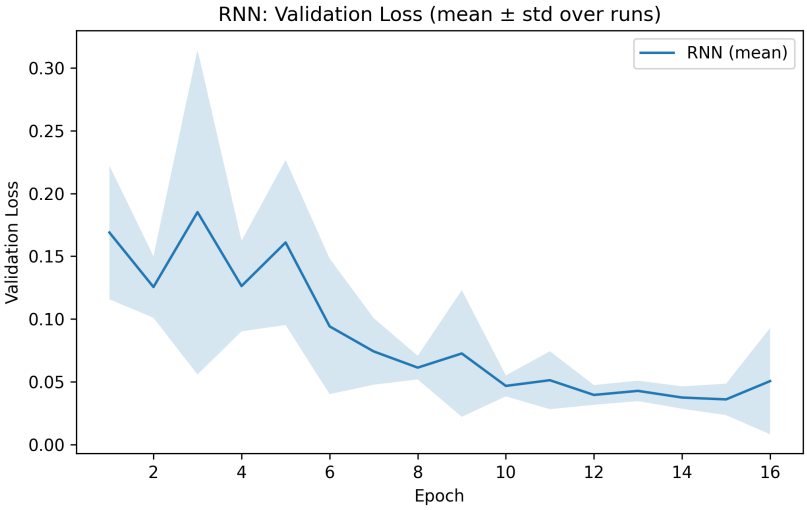


b)

Fig. 6. Accuracy and Loss mean value of LSTM model after 4 rounds

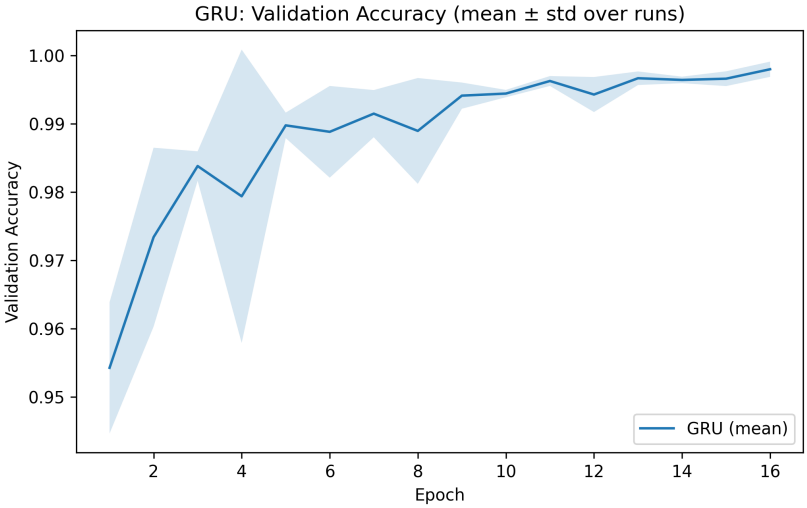


a)

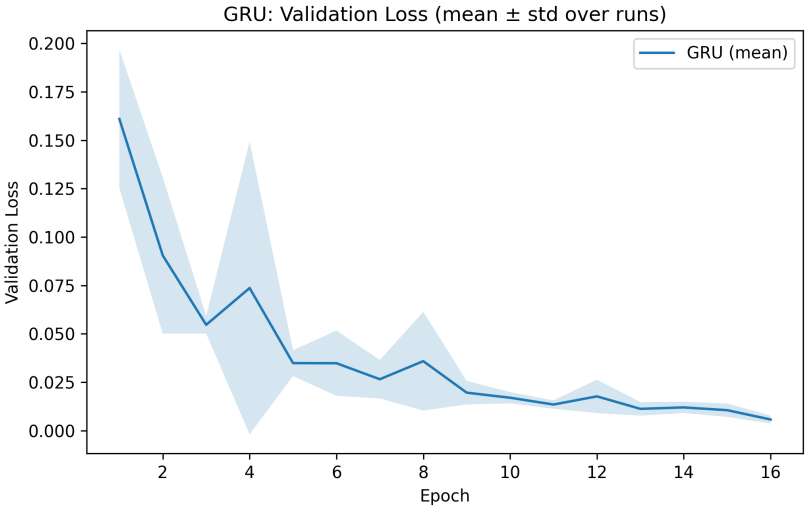


b)

Fig. 7. Accuracy and Loss mean value of RNN model after 4 rounds

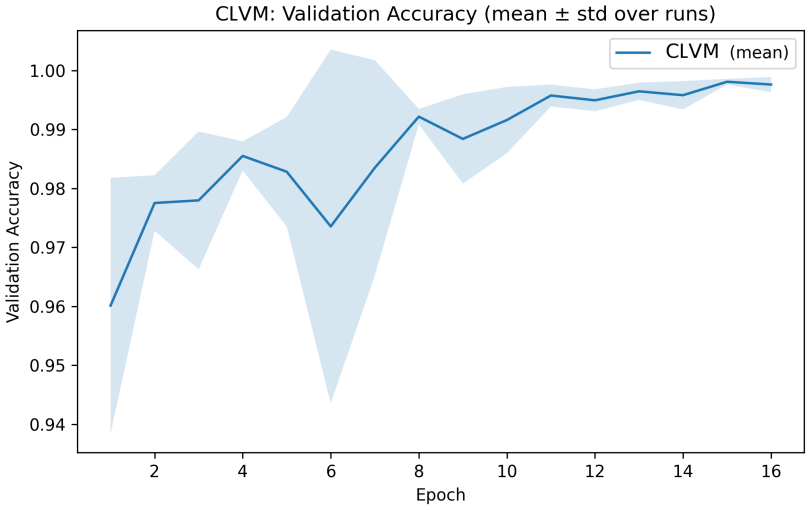


a)

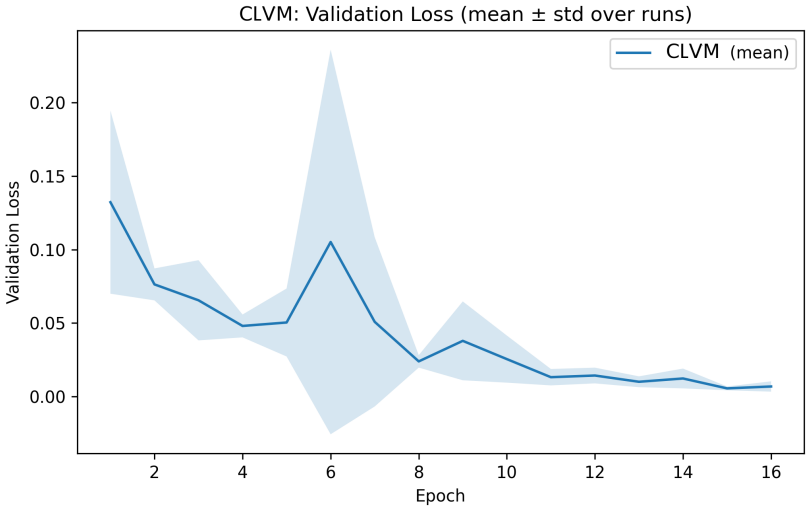


b)

Fig. 8. Accuracy and Loss mean value of GRU model after 4 rounds



a)



b)

Fig. 9. Accuracy and Loss mean value of CLVM model after 4 rounds

5. Conclusion. This paper presented a real-time virtual mouse control system based on hand gesture recognition using the CLVM deep learning architecture. The system uses MediaPipe for real-time hand landmark extraction, CNN to learn spatial features, and LSTM to process motion sequences. The development pipeline encompassed data collection, preprocessing, model design, and performance evaluation. Experimental results show that the proposed model achieved low loss of 0.38, accuracy of 99.88%, and an F1-Score of 99.88%. Compared to reference models (standalone LSTM, GRU, RNN) the system also demonstrates stable processing speed at a higher frame rate. These results demonstrate the effectiveness and reliability of CLVM for real-time hand gesture recognition, even under varying lighting conditions and cluttered backgrounds.

Although the results are promising, the current study is limited to only four basic gestures (Move, Scroll, Pause, Start), serving primarily as proof of concept. Future work will focus on the following directions:

– Expanding the gesture vocabulary to include more complex and practical mouse operations (e.g., single/double-clicking, dragging and dropping, zooming, and multi-finger interactions), thereby improving usability in real-world applications.

– Conducting structured user studies to evaluate experience factors such as intuitiveness, learnability, responsiveness, and potential fatigue during prolonged use, to complement technical performance results.

– Validating the system on standard benchmarks (e.g., SHREC, EgoHands, NVGesture) to enhance external generalizability and enable direct comparison with existing gesture recognition systems.

In summary, CLVM provides a powerful, scalable, and efficient platform for natural human-computer interaction. With planned future developments, it has the potential to evolve from a proof of concept into a versatile and practical alternative to conventional input devices.

References

1. Maslej N., Fattorini L., Perrault R., Gil Y., Parli V., Kariuki N., Capstick E., Reuel A., Brynjolfsson E., Etchemendy J. et al. Artificial intelligence index report. arXiv preprint arXiv:2504.07139. 2025.
2. Asgher U., Ayaz Y., Taiar R. Advances in artificial intelligence (AI) in brain computer interface (BCI) and industry 4.0 for human machine interaction (HMI). Frontiers in Human Neuroscience. 2023. vol. 17. DOI: 10.3389/fnhum.2023.1320536.
3. Sumak B., Brdnik S., Pusnik M. Sensors and artificial intelligence methods and algorithms for human–computer intelligent interaction: A systematic mapping study. Sensors. 2021. vol. 22. no. 1.
4. Mourtzis D., Angelopoulos J., Panopoulos N. The future of the human–machine interface (HMI) in society 5.0. Future Internet. 2023. vol. 15. no. 5.

5. Mukhtar H. Artificial intelligence techniques for human-machine interaction. Artificial Intelligence and Multimodal Signal Processing in Human-Machine Interaction. 2025. pp. 19–42.
6. Shibly K.H., Dey S.K., Islam M.A., Showrav S.I. Design and development of hand gesture based virtual mouse. 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019. pp. 1–5. DOI: 10.1109/ICASERT.2019.8934612.
7. Shrivastava A., Pundir S., Sharma A., Srivastava A., Kumar R., Khan A.K. Control of a virtual system with hand gestures. 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN). IEEE, 2023. pp. 1716–1721.
8. Reddy V.V., Dhyanchand T., Krishna G.V., Maheshwaram S. Virtual mouse control using colored finger tips and hand gesture recognition. IEEE-HYDCON. IEEE, 2020. pp. 1–5. DOI: 10.1109/HYDCON48903.2020.9242677.
9. Kasar M., Kavimandan P., Suryawanshi T., Abbad S. Ai-based real-time hand gesture-controlled virtual mouse. Australian Journal of Electrical and Electronics Engineering. 2024. vol. 21. no. 3. pp. 258–267.
10. Lugaresi C., Tang J., Nash H., McClanahan C., Uboweja E., Hays M., Zhang F., Chang C.-L., Yong M.G., Lee J., et al. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172. 2019.
11. Hung N.V., Loi T.Q., Binh N.H., Nga N.T.T., Huong T.T., Luu D.L. Building an online learning model through a dance recognition video based on deep learning. Informatics and Automation. 2024. vol. 23. no. 1. pp. 101–128.
12. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012. vol. 25.
13. Hochreiter S., Schmidhuber J. Long short-term memory. Neural Computation. 1997. vol. 9. no. 8. pp. 1735–1780.
14. Acharya K. Virtual mouse using hand gestures. Authorea. 2024. DOI: 10.22541/au.173161606.61659157/v1.
15. Beyeler M.. Machine Learning for OpenCV. Birmingham: Packt Publishing, 2017. 382 p.
16. Joshi P., Escriva D.M., Godoy V. OpenCV by example. Packt Publishing Ltd, 2016. 296 p.
17. Howse J. OpenCV computer vision with python. UK: Packt Publishing Birmingham, 2013. vol. 27.
18. Dharavath K., Kumar G.M., Reddy K.R., Reddy M.H. Gesture and voice controlled virtual mouse for elderly people. 2nd International Conference on Networking and Communications (ICNWC). IEEE, 2024. pp. 1–5.
19. Gupta A., Chawla N., Jain R., Thakur N., Devi A. Gesture-based touchless operations: leveraging mediapipe and OpenCV. NEU Journal for Artificial Intelligence and Internet of Things. 2023. vol. 2. no. 1.
20. Bansal B.S., Nailwal D., Bhatt G., Kumar A., Petwal H. Real-time video control via hand and eye movements using opencv and mediapipe. International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit). IEEE, 2024. pp. 270–275.
21. Nandwalkar D.J., Mandal M., Khirari A., Bhalchim T. Control mouse using hand gesture and voice. International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2023. vol. 11. pp. 3261–3268.
22. Hung N.V., Quan N.A., Tan N., Hai T.T., Trung D.K., Nam L.M., Loan B.T., Nga N.T.T. Building predictive smell models for virtual reality environments. Informatics and Automation. 2025. vol. 24. no. 2. pp. 556–582. DOI: 10.15622/ia.24.2.7.
23. Kruk A. The benefits of virtual learning environment (VLE) in teaching ESP. Current nutrition in the humanities. 2022.

24. Hung N., Dat P.T., Tan N., Quan N.A., Trang L., Nam L.M., et al. Heverl–viewport estimation using reinforcement learning for 360-degree video streaming. Informatics and Automation. 2025. vol. 24. no. 1. pp. 302–328.
25. Nguyen H., Dao T.N., Pham N.S., Dang T.L., Nguyen T.D., Truong T.H. An accurate viewport estimation method for 360 video streaming using deep learning. EAI Endorsed Trans. Ind. Networks Intell. Syst. 2022. vol. 9. no. 4.
26. Kumar M., Rathi G., Singh T., NL T. CNN based virtual whiteboard application. Second International Conference on Advances in Information Technology (ICAIT). IEEE, 2024. pp. 1–6.
27. Heemskerk I., Kuiper E., Meijer J. Interactive whiteboard and virtual learning environment combined: Effects on mathematics education. Journal of Computer Assisted Learning. 2014. vol. 30. no. 5. pp. 465–478.
28. Mundargi Z., Das S., Shinde A., Deokar O., Bahirat S., Shetiya D. Hand gesture desktop control with python. 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT). IEEE, 2024. vol. 1. pp. 35–40.
29. Nguyen H.A., Tran T.T., Ho H.Q., Ngo T.D., Vu K.N., Huynh V.L.T. Hand gesture recognition using cvzone. Proceedings of the 9th International Conference on Intelligent Information Technology. 2024. pp. 108–113.
30. Uke S., Shaikh A., Rayate H., Kamble A., Rahane S. Towards touchless interaction: Implementing hand gesture recognition for presentation and media control. International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2025. pp. 1–6. DOI: 10.1109/ESCI63694.2025.10988099.
31. Vasanthagokul S., Kamakshi K.V.G., Mudbhari G., Chithrakumar T. Virtual mouse to enhance user experience and increase accessibility. 4th International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2022. pp. 1266–1271.
32. Naidu P., Muthukumaran N., Chandralekha S., Reddy K.T., Vaishnavi K.S. An analysis on virtual mouse control using human eye. 5th International Conference on Image Processing and Capsule Networks (ICIPCN). IEEE, 2024. pp. 233–237.
33. Praba B.V., Vinothini R., Jayarathna M., Subramani K., Sravanthi P. Virtual AI mouse with biometric authentication. International Conference on Knowledge Engineering and Communication Systems (ICKECS). IEEE, 2024. vol. 1. pp. 1–5. DOI: 10.1109/ICKECS61492.2024.10617264.
34. Karthick S., Dinesh M., Jeffery Dani Raj C., Jayapandian N. Artificial intelligence based enhanced virtual mouse hand gesture tracking using yolo algorithm. IEEE 2nd International Conference on Data, Decision and Systems (ICDDS). IEEE, 2023. pp. 1–6. DOI: 10.1109/ICDDS59137.2023.10434330.
35. Karpov A., Ronzhin A., Kipyatkova I. An assistive Bi-modal user interface integrating multi-channel speech recognition and computer vision. Human-Computer Interaction: Interaction Techniques and Environments (HCI 2011). Lecture Notes in Computer Science. 2011. vol. 6762. pp. 454–463. DOI: 10.1007/978-3-642-21605-3_50.
36. Karpov A., Carbini S., Ronzhin A., Viallet J.E. Comparison of two different similar speech and gestures multimodal interfaces. Proc. of the 16th European Signal Processing Conference (EUSIPCO). 2008. pp. 1–5.
37. Jaimes A., Sebe N. Multimodal human–computer interaction: A survey. Computer Vision and Image Understanding. 2007. vol. 108. no. 1–2. pp. 116–134.
38. Bazarevsky V., Zhang F. On-device, real-time hand tracking with mediapipe. Available at: https://research.google/blog/on-device-real-time-hand-tracking-with-mediapipe/ (accessed 15.01.2026).
39. Manning C.D., Raghavan P., Scutze H. Introduction to Information Retrieval. Cambridge University Press, 2008.

40. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009. vol. 45. no. 4. pp. 427–437.

41. Powers D.M.W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. Journal of Machine Learning Technologies. 2011. vol. 2. no. 1. pp. 37–63.

Nguyen Viet Hung — Ph.D., Lecturer, International Training and Cooperation Institute, East Asia University of Technology. Research interests: multimedia communications, network security, artificial intelligence, traffic engineering in next-generation networks, QoE/QoS guarantee for network services, green networking, applications. The number of publications — 35. hungnv@eaut.edu.vn; Ky Anh, Ha Tinh, Viet Nam; office phone: +84(098)911-2079.

Phi Dinh Huynh — Research assistant, Faculty of Information Technology, East Asia University of Technology. Research interests: nature language processing, machine and deep learning, computer vision. The number of publications — 2. 20222072@eaut.edu.vn; Huong Ngai – Thach That, Ha Noi, Viet Nam; office phone: +84(037)395-0094.

Ma Van Tung — Research assistant, International Training and Cooperation Institute, East Asia University of Technology. Research interests: artificial intelligence, machine and deep learning, virtual reality. The number of publications — 1. 20230240@eaut.edu.vn; Minh Quang, Tuyen Quang, Viet Nam; office phone: +84(092)882-2756.

Nguyen Van Vu — Research assistant, Faculty of Information Technology, East Asia University of Technology. Research interests: machine and deep learning, computer vision. The number of publications — 1. 20222048@eaut.edu.vn; Quoc Oai, Ha Noi, Viet Nam; office phone: +84(038)379-0656.

Nguyen Phu Dat — Research assistant, Faculty of Information Technology, East Asia University of Technology. Research interests: nature language processing, artificial intelligence, computer network. The number of publications — 1. 20220344@eaut.edu.vn; Son Trung – Quoc Oai, Ha Noi, Viet Nam; office phone: +84(096)548-3341.

УДК 006.72 **DOI** 10.15622/ia.25.2.5

Н.В. Хунг, Ф.Д. Хуинь, М.В. Тунг, Н.В. Ву, Н.Ф. Дат

CLVM: ГИБРИДНАЯ МОДЕЛЬ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ БЕСКОНТАКТНОГО УПРАВЛЕНИЯ ВИРТУАЛЬНОЙ МЫШЬЮ

Хунг Н.В., Хуинь Ф.Д., Тунг М.В., Ву Н.В., Дат Н.Ф. **CLVM: гибридная модель глубокого обучения для бесконтактного управления виртуальной мышью.**

Аннотация. В эпоху стремительной цифровой трансформации и растущего распространения искусственного интеллекта обеспечение естественного, непрерывного и бесконтактного человеко-компьютерного взаимодействия приобретает первостепенное значение для различных областей. Данная работа представляет новую модель на базе глубокого обучения для управления виртуальной мышью посредством жестов, получившая название CLVM (CNN-LSTM Virtual Mouse). Разработанная система основывается на гибридной архитектуре, интегрирующей три мощных компонента: (1) MediaPipe – для высокоэффективной детекции ключевых ориентиров кисти в режиме реального времени; (2) сверточную нейронную сеть (CNN) – для извлечения пространственных признаков; (3) сеть долгой краткосрочной памяти (LSTM) – для моделирования временной динамики, что существенно повышает точность и непрерывность распознавания жестов во временной последовательности. В отличие от традиционных подходов, модель CLVM разработана для сохранения высокой производительности в условиях реальной среды, особенно при неравномерном освещении и наличии загроможденного фона. Система характеризуется низкой задержкой и высокой скоростью отклика, а также возможностью эффективного функционирования на устройствах с ограниченными ресурсами, что обуславливает ее пригодность для широкого практического применения. Результаты экспериментов демонстрируют, что CLVM достигает высокой точности (99,88%) при снижении потерь до 0,38, значительно превосходя по эффективности традиционные методы распознавания жестов. Полученные данные подчеркивают потенциал CLVM как надежного, масштабируемого и эффективного решения для организации естественного взаимодействия на основе жестов, представляя собой важный шаг вперед в разработке интеллектуальных, удобных для пользователя интерфейсов для бесконтактного управления.

Ключевые слова: компьютерное зрение, бесконтактный интерфейс, ориентиры кисти, машинное обучение, MediaPipe, виртуальная мышь.

Литература

1. Maslej N., Fattorini L., Perrault R., Gil Y., Parli V., Kariuki N., Capstick E., Reuel A., Brynjolfsson E., Etchemendy J. et al. Artificial intelligence index report. arXiv preprint arXiv:2504.07139. 2025.
2. Asgher U., Ayaz Y., Taiar R. Advances in artificial intelligence (AI) in brain computer interface (BCI) and industry 4.0 for human machine interaction (HMI). Frontiers in Human Neuroscience. 2023. vol. 17. DOI: 10.3389/fnhum.2023.1320536.
3. Sumak B., Brdnik S., Pusnik M. Sensors and artificial intelligence methods and algorithms for human–computer intelligent interaction: A systematic mapping study. Sensors. 2021. vol. 22. no. 1.
4. Mourtzis D., Angelopoulos J., Panopoulos N. The future of the human–machine interface (HMI) in society 5.0. Future Internet. 2023. vol. 15. no. 5.

5. Mukhtar H. Artificial intelligence techniques for human-machine interaction. Artificial Intelligence and Multimodal Signal Processing in Human-Machine Interaction. 2025. pp. 19–42.
6. Shibly K.H., Dey S.K., Islam M.A., Showrav S.I. Design and development of hand gesture based virtual mouse. 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019. pp. 1–5. DOI: 10.1109/ICASERT.2019.8934612.
7. Shrivastava A., Pundir S., Sharma A., Srivastava A., Kumar R., Khan A.K. Control of a virtual system with hand gestures. 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN). IEEE, 2023. pp. 1716–1721.
8. Reddy V.V., Dhyanchand T., Krishna G.V., Maheshwaram S. Virtual mouse control using colored finger tips and hand gesture recognition. IEEE-HYDCON. IEEE, 2020. pp. 1–5. DOI: 10.1109/HYDCON48903.2020.9242677.
9. Kasar M., Kavimandan P., Suryawanshi T., Abbad S. Ai-based real-time hand gesture-controlled virtual mouse. Australian Journal of Electrical and Electronics Engineering. 2024. vol. 21. no. 3. pp. 258–267.
10. Lugaresi C., Tang J., Nash H., McClanahan C., Uboweja E., Hays M., Zhang F., Chang C.-L., Yong M.G., Lee J., et al. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172. 2019.
11. Hung N.V., Loi T.Q., Binh N.H., Nga N.T.T., Huong T.T., Luu D.L. Building an online learning model through a dance recognition video based on deep learning. Informatics and Automation. 2024. vol. 23. no. 1. pp. 101–128.
12. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012. vol. 25.
13. Hochreiter S., Schmidhuber J. Long short-term memory. Neural Computation. 1997. vol. 9. no. 8. pp. 1735–1780.
14. Acharya K. Virtual mouse using hand gestures. Authorea. 2024. DOI: 10.22541/au.173161606.61659157/v1.
15. Beyeler M.. Machine Learning for OpenCV. Birmingham: Packt Publishing, 2017. 382 p.
16. Joshi P., Escriva D.M., Godoy V. OpenCV by example. Packt Publishing Ltd, 2016. 296 p.
17. Howse J. OpenCV computer vision with python. UK: Packt Publishing Birmingham, 2013. vol. 27.
18. Dharavath K., Kumar G.M., Reddy K.R., Reddy M.H. Gesture and voice controlled virtual mouse for elderly people. 2nd International Conference on Networking and Communications (ICNWC). IEEE, 2024. pp. 1–5.
19. Gupta A., Chawla N., Jain R., Thakur N., Devi A. Gesture-based touchless operations: leveraging mediapipe and OpenCV. NEU Journal for Artificial Intelligence and Internet of Things. 2023. vol. 2. no. 1.
20. Bansal B.S., Nailwal D., Bhatt G., Kumar A., Petwal H. Real-time video control via hand and eye movements using opencv and mediapipe. International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit). IEEE, 2024. pp. 270–275.
21. Nandwalkar D.J., Mandal M., Khirari A., Bhalchim T. Control mouse using hand gesture and voice. International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2023. vol. 11. pp. 3261–3268.
22. Hung N.V., Quan N.A., Tan N., Hai T.T., Trung D.K., Nam L.M., Loan B.T., Nga N.T.T. Building predictive smell models for virtual reality environments. Informatics and Automation. 2025. vol. 24. no. 2. pp. 556–582. DOI: 10.15622/ia.24.2.7.
23. Kruk A. The benefits of virtual learning environment (VLE) in teaching ESP. Current nutrition in the humanities. 2022.

24. Hung N., Dat P.T., Tan N., Quan N.A., Trang L., Nam L.M., et al. Heverl–viewport estimation using reinforcement learning for 360-degree video streaming. Informatics and Automation. 2025. vol. 24. no. 1. pp. 302–328.
25. Nguyen H., Dao T.N., Pham N.S., Dang T.L., Nguyen T.D., Truong T.H. An accurate viewport estimation method for 360 video streaming using deep learning. EAI Endorsed Trans. Ind. Networks Intell. Syst. 2022. vol. 9. no. 4.
26. Kumar M., Rathi G., Singh T., NL T. CNN based virtual whiteboard application. Second International Conference on Advances in Information Technology (ICAIT). IEEE, 2024. pp. 1–6.
27. Heemskerk I., Kuiper E., Meijer J. Interactive whiteboard and virtual learning environment combined: Effects on mathematics education. Journal of Computer Assisted Learning. 2014. vol. 30. no. 5. pp. 465–478.
28. Mundargi Z., Das S., Shinde A., Deokar O., Bahirat S., Shetiya D. Hand gesture desktop control with python. 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT). IEEE, 2024. vol. 1. pp. 35–40.
29. Nguyen H.A., Tran T.T., Ho H.Q., Ngo T.D., Vu K.N., Huynh V.L.T. Hand gesture recognition using cvzone. Proceedings of the 9th International Conference on Intelligent Information Technology. 2024. pp. 108–113.
30. Uke S., Shaikh A., Rayate H., Kamble A., Rahane S. Towards touchless interaction: Implementing hand gesture recognition for presentation and media control. International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2025. pp. 1–6. DOI: 10.1109/ESCI63694.2025.10988099.
31. Vasanthagokul S., Kamakshi K.V.G., Mudbhari G., Chithrakumar T. Virtual mouse to enhance user experience and increase accessibility. 4th International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2022. pp. 1266–1271.
32. Naidu P., Muthukumaran N., Chandralekha S., Reddy K.T., Vaishnavi K.S. An analysis on virtual mouse control using human eye. 5th International Conference on Image Processing and Capsule Networks (ICIPCN). IEEE, 2024. pp. 233–237.
33. Praba B.V., Vinothini R., Jayarathna M., Subramani K., Sravanthi P. Virtual AI mouse with biometric authentication. International Conference on Knowledge Engineering and Communication Systems (ICKECS). IEEE, 2024. vol. 1. pp. 1–5. DOI: 10.1109/ICKECS61492.2024.10617264.
34. Karthick S., Dinesh M., Jeffery Dani Raj C., Jayapandian N. Artificial intelligence based enhanced virtual mouse hand gesture tracking using yolo algorithm. IEEE 2nd International Conference on Data, Decision and Systems (ICDDS). IEEE, 2023. pp. 1–6. DOI: 10.1109/ICDDS59137.2023.10434330.
35. Karpov A., Ronzhin A., Kipyatkova I. An assistive Bi-modal user interface integrating multi-channel speech recognition and computer vision. Human-Computer Interaction: Interaction Techniques and Environments (HCI 2011). Lecture Notes in Computer Science. 2011. vol. 6762. pp. 454–463. DOI: 10.1007/978-3-642-21605-3_50.
36. Karpov A., Carbini S., Ronzhin A., Viallet J.E. Comparison of two different similar speech and gestures multimodal interfaces. Proc. of the 16th European Signal Processing Conference (EUSIPCO). 2008. pp. 1–5.
37. Jaimes A., Sebe N. Multimodal human–computer interaction: A survey. Computer Vision and Image Understanding. 2007. vol. 108. no. 1–2. pp. 116–134.
38. Bazarevsky V., Zhang F. On-device, real-time hand tracking with mediapipe. Available at: https://research.google/blog/on-device-real-time-hand-tracking-with-mediapipe/ (accessed 15.01.2026).
39. Manning C.D., Raghavan P., Scutze H. Introduction to Information Retrieval. Cambridge University Press, 2008.

40. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009. vol. 45. no. 4. pp. 427–437.
41. Powers D.M.W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. Journal of Machine Learning Technologies. 2011. vol. 2. no. 1. pp. 37–63.

Хунг Нгуен Вьет — Ph.D., преподаватель, международный институт подготовки кадров и сотрудничества, Восточноазиатский технологический университет. Область научных интересов: мультимедийные коммуникации, сетевая безопасность, искусственный интеллект, управление трафиком в сетях следующего поколения, гарантия качества обслуживания (QoE/QoS) для сетевых сервисов, экологичные сети, приложения. Число научных публикаций — 35. hungnv@eaut.edu.vn; Ки Ань, Хатинь, Вьетнам; р.т.: +84(098)911-2079.

Хуинь Фи Динь — научный сотрудник, факультет информационных технологий, Восточноазиатский технологический университет. Область научных интересов: обработка естественного языка, машинное и глубокое обучение, компьютерное зрение. Число научных публикаций — 2. 20222072@eaut.edu.vn; Хуонг Нгай – Тхач Тхат, Ханой, Вьетнам; р.т.: +84(037)395-0094.

Тунг Ма Ван — научный сотрудник, международный институт обучения и сотрудничества, Восточноазиатский технологический университет. Область научных интересов: искусственный интеллект, машинное и глубокое обучение, виртуальная реальность. Число научных публикаций — 1. 20230240@eaut.edu.vn; Минь Куанг, Туенкуанг, Вьетнам; р.т.: +84(092)882-2756.

Ву Нгуен Ван — научный сотрудник, факультет информационных технологий, Восточноазиатский технологический университет. Область научных интересов: машинное и глубокое обучение, компьютерное зрение. Число научных публикаций — 1. 20222048@eaut.edu.vn; Куок Оай, Ханой, Вьетнам; р.т.: +84(038)379-0656.

Дат Нгуен Фу — научный сотрудник, факультет информационных технологий, Восточноазиатский технологический университет. Область научных интересов: обработка естественного языка, искусственный интеллект, компьютерные сети. Число научных публикаций — 1. 20220344@eaut.edu.vn; Сон Чунг – Куок Оай, Ханой, Вьетнам; р.т.: +84(096)548-3341.

DOI 10.15622/ia.25.2.6

M.A. ATMANE, H. ZATLA, B. TOLBI, S.F. NOUAR, M. BOUHAMAMA

INTELLIGENT FAULT DETECTION AND ISOLATION BASED ON NARX NEURAL NETWORKS

Atmane M.A., Zatla H., Tolbi B., Nouar S.F., Bouhamama M. **Intelligent Fault Detection and Isolation Based on NARX Neural Networks.**

Abstract. Intelligent systems have become an essential component of modern technological landscapes. Their reliability is a major concern, as faults can lead to catastrophic consequences for their behavior and overall performance, making Fault Detection and Isolation (FDI) a critical task for such systems. The complex nonlinear dynamics involved in their operation make this task more challenging. In this context, this paper proposes an intelligent fault detection and isolation methodology, using an incubator system as a representative case study. The proposed method employs a Nonlinear Autoregressive Exogenous (NARX) neural network in a parallel structure to model the complex nonlinear system dynamics. Different implementations of the NARX model, based on Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Elman networks, were compared to assess their modeling performance and determine the best-performing model. The differences between these predictions and the actual outputs are referred to as residuals. For fault classification, a comparative study was conducted among five machine learning (ML) methods: Multilayer Perceptron, Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Linear Discriminant Analysis (LDA). These methods analyze the residuals to isolate the specific fault from predefined potential faults, including both actuator and sensor faults. The results demonstrate the effectiveness of the intelligent FDI methodology presented in this work. The NARX models proved effective with high modeling performance, particularly the MLP-NARX, which outperformed the other models. The comparative study of classifiers highlights the performance differences among the five methods, with the MLP classifier achieving the best results across all metrics. This confirms its suitability for practical FDI applications, due to its strong ability to capture complex nonlinear relationships in the data.

Keywords: fault detection and isolation, incubator, multilayer perceptron, NARX modeling, residual generation and evaluation, recurrent neural networks, machine learning classifiers, intelligent systems.

1. Introduction. Intelligent systems are increasingly prevalent across modern technological applications. These systems perceive their environment, act rationally in it, and interact with agents, with the objective of maximizing operational success. Such systems are exposed to faults, which disrupt this process and can lead to significant consequences. In this context, fault detection and isolation represent a critical task for intelligent systems, enhancing reliability and minimizing operational costs. Fault detection refers to the process of determining whether a fault has occurred and when, while fault isolation consists of identifying which specific fault from a predefined set has occurred. A representative example of such systems is egg incubators [1], where maintaining precise conditions of temperature and humidity is crucial for proper embryo development. Any

fault can compromise these conditions, resulting in serious consequences. A fault in the humidifier or heating system, for example, can cause rapid fluctuations in the internal environment. The criticality of such faults stems from the significant influence of temperature and humidity on hatchability rates and other key parameters, as evidenced by studies [2, 3]. Such failures also have economic repercussions. A lower hatching rate means reduced production and revenue losses for producers. For consumers, this could lead to higher prices for eggs and related products.

In this paper, an intelligent Fault Detection and Isolation (FDI) methodology is proposed, with an incubator system as a representative case study. The complex nonlinear dynamics of the system were captured through a Nonlinear Autoregressive Exogenous (NARX) model in a parallel structure, for which we compared four types of neural networks implementing it: Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Elman. The fault classification was performed using five different machine learning methods: MLP, Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Linear Discriminant Analysis (LDA). The main contributions of this work are: (1) the proposal and validation of an intelligent FDI methodology that avoids reliance on a mathematical model; (2) a comparison of different types of neural networks implementing the NARX model to determine the one best suited to capture the complex nonlinear dynamics of the system; (3) a comparative study of ML classifiers to improve fault classification performance; (4) the introduction of FDI to the rarely explored field of the poultry industry via application to an incubator system.

This paper is organized as follows. Section 2 presents a review of related work. In Section 3 the design and implementation of the incubator is described, while Section 4 presents the methodology of FDI, which includes residual generation and evaluation. Section 5 outlines the research stages, providing an overall view of the research process. Section 6 details the application of this methodology to our case, including the modeling and classification parts, as well as the introduction of faults into the system. Section 7 presents the results, along with their analysis.

2. Related work. Numerous approaches have been developed for FDI across various applications [4]. These approaches can be classified into two main categories: model-based and intelligent methods.

The first category involves model-based methods [5, 6], which rely on a mathematical model to represent the dynamic behavior of the fault-free system. The difference between the model's predicted output and the actual output measurements generates residuals. These residuals are analyzed to

extract relevant information about faults. This process corresponds to what is known as residual generation and residual evaluation. The main model-based techniques used for FDI are based on observers [7, 8], which estimate the internal states of the system based on measurable outputs, and Kalman filters [9, 10], which rely on a system model to generate optimal state estimates and detect anomalies in system behavior.

The main obstacle in using model-based methods lies in the difficulty of obtaining a precise mathematical model of the system, which is generally a complex and costly task. This challenge is particularly pronounced for systems exhibiting complex nonlinear behavior, such as egg incubators, which limits the practical applicability of model-based FDI methods for these systems.

On the other hand, intelligent methods do not depend on a mathematical model, as they are trained using only the available input-output data from the system or using system knowledge. This characteristic makes them particularly suitable for FDI tasks in egg incubator systems. Across different application domains, various intelligent methods have been applied to FDI in the literature, such as Artificial Neural Networks (ANNs) [11 – 15], fuzzy logic systems [16, 17], and SVM [18, 19].

In addition to these approaches, hybrid methods have been proposed, which combine intelligent and model-based techniques. In such cases, intelligent methods are often used as classifiers rather than dynamic models; they analyze the residuals generated by the model-based methods [20, 21].

ANNs are one of the most popular Artificial Intelligence tools used in FDI. Among the advantages of these methods are their capacity to extract generalized knowledge from the available measurement data, their ability to operate autonomously and their capacity to process data precisely even under conditions of uncertainty. Among the various types of neural networks (NNs) used in FDI, such as Radial Basis Function (RBF) networks [22, 23], Convolutional Neural Networks (CNNs) [24, 25], and Recurrent Neural Networks (RNNs) [26 – 28], one of the most widely used is the MLP [29, 30], due to its simplicity and wide applicability.

Considering their capacity to capture complex, nonlinear relationships in system behavior, NNs have been proposed as dynamic models for FDI applications [31 – 33]. Among the various NN architectures, the NARX stands out as one of the most suitable and widely used for such tasks, as it explicitly incorporates past inputs and outputs, enabling it to model the temporal dependencies inherent to dynamic systems. In [34], the authors applied a NARX neural network to model a distillation column for a fault detection task, highlighting its ability to capture the nonlinearity of the process and model the system dynamics

based on time-series data. Such capabilities align with the complex nonlinear dynamics of egg incubators, making NARX neural networks highly relevant for FDI applications in these systems. In [35], the authors demonstrated their suitability for room temperature and relative humidity prediction in indoor environments. NARX neural networks have demonstrated higher prediction accuracy compared to traditional linear models [36, 37], such as ARX and ARMAX. The performance of these models is limited when dealing with complex nonlinear relationships [38], as they are based on the assumption of linear dependencies among system variables. In [39], the authors demonstrated the limitations of ARX models in capturing the nonlinear dynamics of hydrological variables, whereas the NARX neural network proved more suitable and achieved superior performance. These limitations of linear models were further highlighted in [40], where the ARX model was shown to be limited in capturing heat transfer nonlinearities for indoor temperature prediction.

In the literature, several regression algorithms have been employed to implement the NARX model [41]. In [42], the authors proposed a method for predicting the Mach number based on an Elman-NARX model. In [43], a tide level prediction method was presented, with an approach based on NARX models implemented by three types of recurrent neural networks: LSTM, GRU, and bidirectional long short-term memory (BiLSTM). Another work [44] compared the MLP-NARX and LSTM-NARX models for the velocity prediction of a pipeline inspection gauge.

In addition to modeling tasks, the ability of NNs to learn complex patterns in input-output data makes them also a relevant choice for classification. They can handle nonlinear relationships and learn directly from labeled data. Among the most commonly used NN architectures for classification tasks are the MLP [45, 46] and the RBF network [47, 48].

Another popular approach for classification within intelligent methods is SVM [49, 50]. SVM is a powerful technique based on statistical learning theory. It is particularly effective in high-dimensional spaces and demonstrates good generalization capabilities. However, its performance can be affected by noisy or overlapping classes, and it heavily depends on the choice of the kernel function and tuning parameters.

XGBoost is also widely applied for classification tasks. This algorithm has been reported to achieve high accuracy and precision [51, 52], but its complexity makes model interpretation challenging.

LDA is another commonly employed method for classification tasks [53, 54], as it focuses on maximizing the separation between classes. However, it relies on strong statistical assumptions about the data

distribution, which may limit its effectiveness when these assumptions are not met.

Another method commonly used for classification is KNN [55, 56]. It is simple to understand and implement, as it is essentially a distance-based algorithm that makes predictions based on the nearest neighbors. However, noisy data can significantly affect its performance.

The review of existing literature highlights a limited number of studies applying FDI methods to egg incubators, despite the severe consequences that faults can have on such systems. Some works focused solely on monitoring temperature and humidity parameters, displaying sensor readings so that the user can follow the proper functioning of the incubator [57 – 59]. Most existing studies are limited to basic fault detection based on thresholding, where faults are detected when specific physical parameters exceed predefined limits [60 – 62]. These approaches can only indicate the presence or absence of a fault and perform poorly in complex or varying operating conditions. Other works relied on rule-based reasoning [63], which requires an extensive database of rules. However, building such a database is time-consuming, requires detailed knowledge of the system, and offers limited adaptability to changes in the system or its operating conditions.

This study proposes an intelligent FDI methodology that involves two key components: (1) modeling the complex nonlinear system dynamics using a NARX model for which different neural network implementations are compared, and (2) a comparative study of different machine learning classifiers. The aim is to achieve improved fault isolation and, consequently, better system performance. In light of the very limited prior work on FDI for egg incubators, mostly restricted to parameter monitoring or basic fault detection, the proposed methodology is applied to this system, requiring neither a mathematical model nor expert-defined rules.

3. System description. This section focuses on the design and implementation of a custom incubator system for controlling temperature and humidity in a closed environment.

The system is composed of two chambers. The first chamber contains the humidity source, which consists of an ultrasonic humidifier submerged in water and a speed-adjustable fan. The fan pushes the moisture generated by the humidifier into the second chamber, which contains an adjustable power lamp acting as a heat source. This chamber also includes a DHT22 sensor to measure both temperature and humidity. Data are acquired every 5 seconds. The previously described system is schematized in Figure 1.

The entire system is controlled by an Arduino Mega board. The fan speed is regulated through a 5V pulse-width modulation (PWM) signal, with values ranging from 0 to 255. The lamp is powered by a variable 220V AC voltage and is controlled by a 5V PWM signal from the Arduino. This signal is sent to a dimmer, which correspondingly adjusts the 220V AC voltage supplied to the lamp.

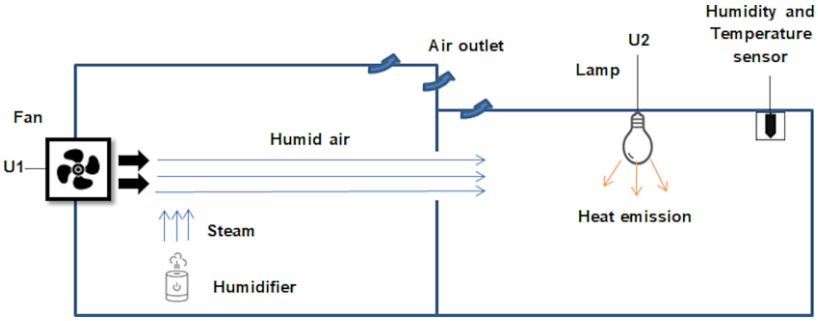


Fig. 1. Schematic representation of the incubator design

To illustrate the practical implementation of the proposed system, Figure 2 presents real images of the custom incubator. Given its dynamic nature, the parameters of the system are inherently temporal; therefore, the temperature and humidity measurements form time-series data. The system dynamics are characterized by nonlinear dependencies between the system's parameters, which arise from fundamental thermodynamic relationships. Temperature variations cause relative humidity to respond nonlinearly, as the amount of water vapor the air can hold changes according to the Clausius-Clapeyron relation. At the same time, the water vapor content influences temperature dynamics by modifying the heat capacity of the air in the incubator [64]. The air temperature also varies nonlinearly with the applied lamp voltage due to Joule heating.

The temperature output y_1 is approximated in Equation (1) as a function of its past values, the past values of the humidity output y_2 and the past values of the control inputs u_1 for the fan and u_2 for the lamp. f_1 is the describing function of the temperature output.

$$y_1(t) = f_1(y_1(t-1), y_1(t-2), \dots, y_2(t-1), y_2(t-2), \dots, u_1(t-1), u_1(t-2), \dots, u_2(t-1), u_2(t-2), \dots). \quad (1)$$

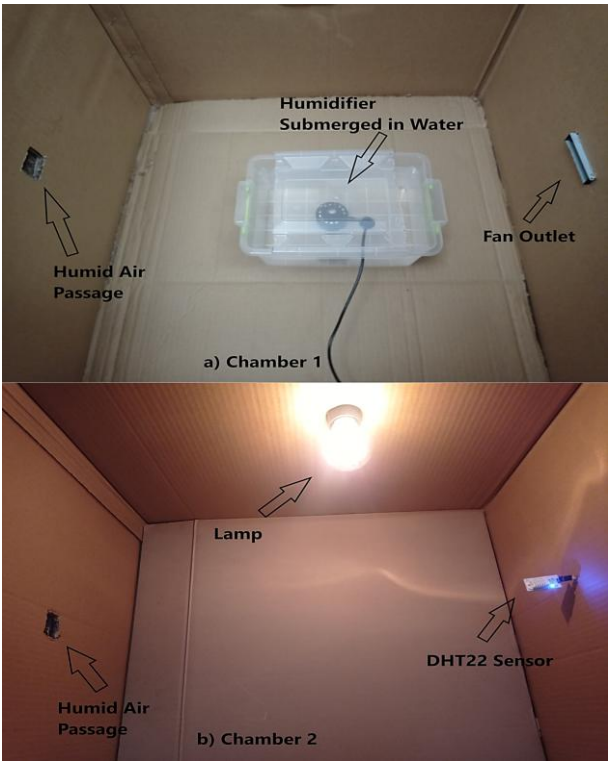


Fig. 2. Real images of the custom incubator: a) chamber 1, b) chamber 2

The notation $(t-1)$ simplifies the representation of the system's previous value at the preceding sample, indicating that the current value depends on its immediate past value.

The same principle applies to humidity, as shown in Equation (2), which is defined over the same set of variables as in Equation (1), where f_2 is the describing function of the humidity output.

$$y_2(t) = f_2(y_2(t-1), y_2(t-2), \dots, y_1(t-1), y_1(t-2), \dots, u_1(t-1), u_1(t-2), \dots, u_2(t-1), u_2(t-2), \dots). \quad (2)$$

4. FDI methodology. The fault detection and isolation process comprises two main tasks: residual generation and residual evaluation.

The overall FDI methodology is presented in Figure 3. Firstly, an NN model is trained using data collected from the fault-free system, known as the healthy condition. Then, the model is used in parallel with the actual

system to predict the outputs. The difference between the system outputs and the model predictions forms the residual signal. This constitutes the residual generation stage. Thus, under fault-free conditions, the residual consists solely of modeling error caused by noise and model-plant mismatch. When a fault occurs, the system output is affected and deviates from its nominal values, while the model prediction remains unaffected by the fault. Consequently, the residual has a significant deviation from zero, which is caused by faults.

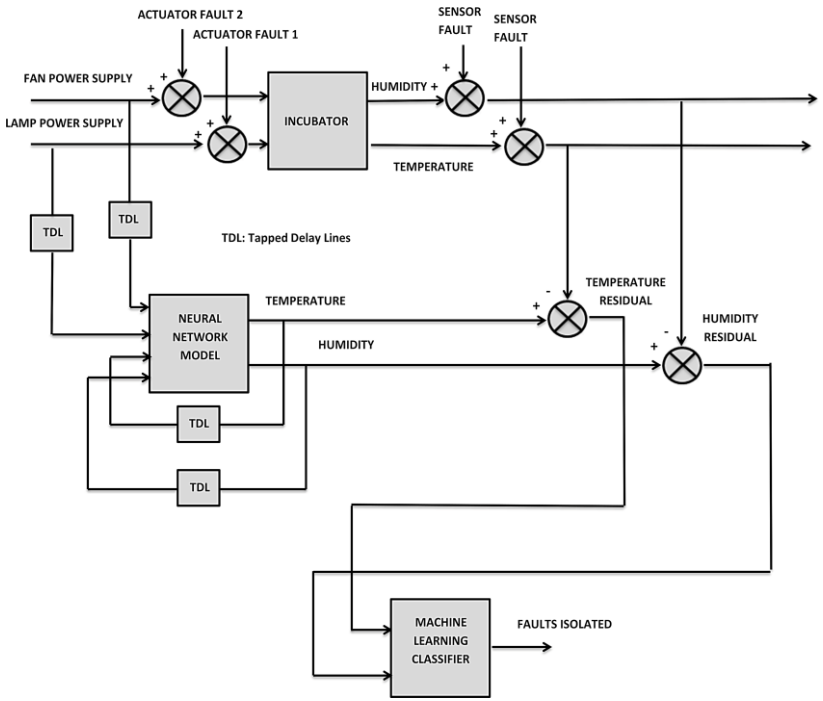


Fig. 3. Schematic representation of the FDI methodology

The model used is a NARX network, which is capable of capturing complex nonlinear dynamics. A comparative study is conducted among four different neural network architectures that implement this model: MLP, Elman, LSTM, and GRU. The best-performing architecture is then employed for residual generation.

The first type of network considered is the MLP [65]. It is a feedforward neural network composed of an input layer, one or more hidden layers, and an output layer. It uses nonlinear activation functions to

approximate complex relationships. The general formulation of the MLP can be expressed as:

$$y(t) = f_y(W_2 f_h(W_1 x(t) + b_1) + b_2), \tag{3}$$

where x is the input vector; y is the output vector; W_1 and W_2 are the learnable input-hidden and hidden-output weight matrices, respectively; b_1 and b_2 are the corresponding bias vectors; f_h is the nonlinear activation function of the hidden layer; and f_y is the output activation function.

The Elman network is also considered [66]. It is a type of recurrent neural network that includes additional feedback connections from the hidden layer output to the input layer. The recurrent units transfer the previous hidden state back to the input, acting as a one-step time delay. The mathematical formulation of the Elman network is given by:

$$h(t) = f_h(W_h[h(t-1), x(t)] + b_h), \tag{4}$$

where h is the hidden state; W_h is the learnable weight matrix of the hidden state; and b_h is the corresponding bias vector.

The LSTM is also employed [67]. It is a type of recurrent neural network capable of learning long-term dependencies through the introduction of a memory cell that can preserve information over long time intervals. Three different gates regulate the information flow within an LSTM cell: the input gate, the forget gate, and the output gate, which together determine what information should be added, retained, or discarded from the cell state. The mathematical representation of the LSTM can be formulated as:

$$\begin{gathered} i(t) = \sigma(W_i[h(t-1), x(t)] + b_i), \\ f(t) = \sigma\big(W_f[h(t-1), x(t)] + b_f\big), \\ g(t) = \tanh\,(W_g[h(t-1), x(t)] + b_g), \\ o(t) = \sigma(W_o[h(t-1), x(t)] + b_o), \\ c(t) = f(t) * c(t-1) + i(t) * g(t), \\ h(t) = o(t) * \tanh\big(c(t)\big), \end{gathered} \tag{5}$$

where c is the new cell state; g is the candidate cell state; σ is the sigmoid function; $*$ is the Hadamard product; i, f, and o are the input, forget, and output gates, respectively; W_i, W_f, W_g, and W_o are the learnable weight matrices of the input gate, the forget gate, the candidate cell state, and the

output gate, respectively; b_i, b_f, b_g, and b_o are the corresponding bias vectors.

The GRU network is also used [68]. It is a recurrent neural network adapted from the LSTM, which uses only two gates: an update gate and a reset gate. The input and forget gates of the LSTM are merged into an update gate. The GRU can be mathematically represented as:

$$\begin{aligned} r(t) &= \sigma(W_r[h(t-1), x(t)] + b_r), \\ z(t) &= \sigma(W_z[h(t-1), x(t)] + b_z), \\ n(t) &= \tanh(W_n[r(t) * h(t-1), x(t)] + b_n), \\ h(t) &= \big(1 - z(t)\big) * n(t) + z(t) * h(t-1), \end{aligned} \tag{6}$$

where n is the candidate hidden state; r and z are the reset and update gates, respectively; W_r, W_z, and W_n are the learnable weight matrices of the reset gate, the update gate, and the candidate hidden state, respectively; b_r, b_z, and b_n are the corresponding bias vectors.

To perform residual evaluation, different machine learning methods are used.

An MLP is employed as a classifier for residual evaluation. Faults affect the residual vector in different ways, making it a valuable source of information about all the faults. As a result, the residual vector serves as the input to the neural classifier, which is trained as follows. Residual data are collected for all system states, encompassing both the healthy condition and every faulty state. These data are then introduced as inputs to the classifier. For each state, the classifier's target output vector is set to 0, except for the element corresponding to that particular state, which is set to 1. Thus, when the neural classifier is tested to isolate faults with residual inputs, one of its outputs activates to '1' to indicate that the system is in the state associated with that output. This characteristic allows for clear fault isolation, given that all possible faults exhibit distinct characteristics and that they occur separately.

The other methods are trained similarly, using the same residual vectors under known system states as inputs, but with a unique label assigned to each state as the output.

Fault isolation is also performed using an SVM. An SVM is a supervised learning technique designed to classify data into different classes by constructing hyperplanes in a high-dimensional space. During fault isolation, the SVM identifies the system state based on the position of the residual in input relative to the hyperplanes and assigns the corresponding label in output.

XGBoost is also used. It is a machine learning method that utilizes ensembles, which combine multiple weak learners, typically decision trees, to build a strong predictive model.

LDA is another applied method. It is a linear technique that performs dimensionality reduction to separate multiple classes, thereby enabling fault isolation.

The KNN method is also employed. It assigns a class label to a sample by analyzing the K nearest neighbors in the feature space and selecting the most frequent class among them.

The performance of these methods in fault isolation is compared and evaluated.

5. Research stages. In the present study, the proposed FDI methodology was applied through a sequence of clearly defined stages. This section briefly outlines these steps to provide a clear understanding of the overall research process.

– **System modeling.** Data were collected from the incubator in the healthy state. A set of different amplitude steps was generated for each of the two inputs to obtain a representative dataset. Different NARX models, each based on a distinct neural network type, were then trained on these data to capture the system's dynamic behavior in the healthy state, thereby enabling residual generation. A linear ARX model was also estimated for comparison with these NARX neural networks, in order to validate their use within the proposed approach.

– **Simulating faults.** To acquire system data under faulty conditions, controlled fault scenarios were created by manually introducing faults into the system. Actuator faults were applied at the control level, while sensor faults were introduced at the output level.

– **Fault classification.** Residual data were collected from each system state, including both healthy and faulty conditions. Different machine learning classifiers were trained on these data to enable fault isolation.

– **Model evaluation.** The trained models were assessed by computing regression metrics, namely. the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), to compare their performance and identify the best-performing model.

– **Classifier evaluation.** The trained classifiers were tested on residual datasets representing healthy and faulty system states. Their performance was assessed through a comparative study based on accuracy, precision, recall, and F1-score, with the aim of identifying the most effective method for fault isolation.

6. Application. This section details the application of the proposed methodology to the egg incubator system.

6.1. Incubator modeling. The capacity to understand and predict complex system behavior has improved considerably with the introduction of NNs for nonlinear dynamic modeling. Their proven effectiveness in capturing complex nonlinear dynamics of systems has made NARX models an ideal choice for this modeling task. The NARX architectures include two main structures: parallel and series-parallel.

In the series-parallel structure referred to as open-loop architecture, the output of the process serves as an input to the NN. The network training is significantly simpler in the open-loop architecture. After the open-loop training process is completed, the network can perform one-step-ahead prediction. However, the objective is system modeling, which requires multi-step-ahead prediction. This is only possible in the parallel structure referred to as the closed-loop architecture which uses the model's own output, instead of the process output, as part of the inputs. However, a network trained only in open-loop often performs poorly when switched to closed-loop operation. For this reason, the networks were trained directly in the closed-loop configuration. Although more challenging, this approach ensures good performance in the parallel structure and enables accurate multi-step-ahead predictions.

Different implementations of the NARX model were realized using four types of neural networks: MLP, LSTM, GRU, and Elman, serving as nonlinear function approximators.

The incubator system has two input variables – the power supplied to the fan and the lamp – and two output variables: temperature and humidity. Creating an informative training dataset is one of the most critical tasks in modeling, as it plays a crucial role in the accuracy of the neural network. To obtain a representative dataset, a set of different amplitude steps was generated for each of the two system inputs: the power supply for the fan and the lamp. Each step had a duration of 20 minutes. The excitation signals, which correspond to PWM signals controlling the two actuators, were bounded between 0 and 125 for the lamp and between 0 and 255 for the fan. Each signal had a total duration of 10 hours. The excitation signals are shown in Figure 4. The dataset presents several challenges, including missing values and erroneous zero values. These challenges are especially significant given the time-series nature of the data, as they disrupt continuity, leading to potentially inaccurate predictions. Missing values were imputed using linear interpolation, which assumes a linear trend between the two known values surrounding the missing entry. Zero values were treated similarly. This approach restores data continuity, aligning the

interpolated values with the context provided by the surrounding data points.

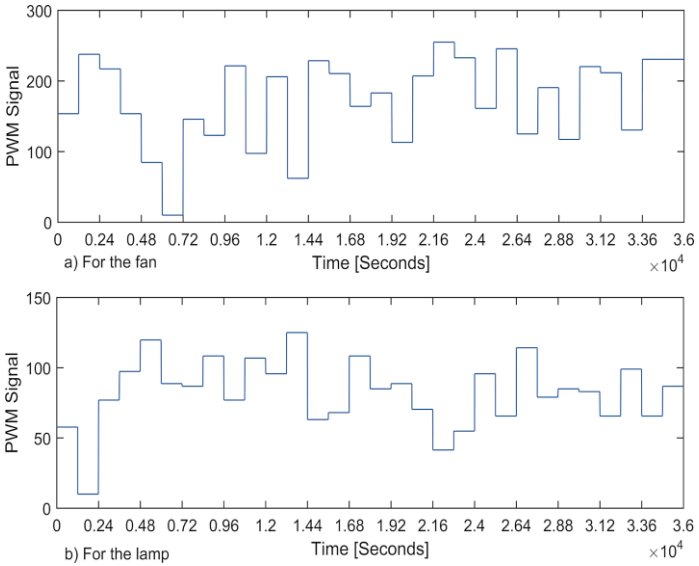


Fig. 4. Excitation signals: a) for the fan, b) for the lamp

The data were divided as follows:

– **Training:** 70%.
– **Validation:** 15%.
– **Testing:** 15%.

Due to the interconnected nature of the system, our NARX models were configured as multi-input multi-output (MIMO) systems. Past input and output values were introduced using Tapped Delay Lines (TDLs), which enable the model to capture the temporal dependencies present in the system's time-series data. We selected the output delays $d_1 = d_2 = 6$ and input delays $d_3 = d_4 = 6$ for the system modeling. These delay values correspond to the number of past samples used for prediction.

The predicted outputs y_{1p} and y_{2p} are given by Equations (7) and (8), where f_{1p} and f_{2p} are the neural network's input-output relation for temperature and humidity output, respectively; d_1 and d_2 denote the output delays; d_3 and d_4 are the input delays; and u_1 and u_2 represent the control inputs for the fan and lamp.

$$y_{1p}(t) = f_{1p}(y_{1p}(t-d_1:t-1), y_{2p}(t-d_2:t-1), u_1(t-d_3:t-1), u_2(t-d_4:t-1)). \quad (7)$$

$$y_{2p}(t) = f_{2p}(y_{2p}(t-d_2:t-1), y_{1p}(t-d_1:t-1), u_1(t-d_3:t-1), u_2(t-d_4:t-1)). \quad (8)$$

Configuring a neural network involves several decisions, including the number of hidden layers and neurons, the choice of activation functions, the optimization algorithm, the loss function, and the number of epochs. These parameters significantly influence the network's performance.

For the MLP network, the following configuration was used:

– **Number of hidden layers and neurons:** 1 hidden layer with 10 neurons.

– **Activation functions:** Sigmoid hidden layer, Linear output layer.

– **Number of epochs:** 1000 epochs.

– **Optimization algorithm:** Bayesian regularization backpropagation algorithm [69 – 71].

For the Elman network, the following configuration was used:

– **Number of hidden layers and neurons:** 1 hidden layer with 45 neurons.

– **Activation functions:** Hyperbolic tangent hidden layer, Linear output layer.

– **Number of epochs:** 201 epochs.

– **Optimization algorithm:** Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimization algorithm [72].

For the LSTM network, the following configuration was used:

– **Number of hidden layers and neurons:** 1 hidden layer with 45 neurons.

– **Activation functions:** Sigmoid gate activation, Hyperbolic tangent state activation, Linear output layer.

– **Number of epochs:** 247 epochs.

– **Optimization algorithm:** L-BFGS optimization algorithm**.**

For the GRU network, the following configuration was used:

– **Number of hidden layers and neurons:** 1 hidden layer with 50 neurons.

– **Activation functions:** Sigmoid gate activation, Hyperbolic tangent state activation, Linear output layer.

– **Number of epochs:** 139 epochs.

– **Optimization algorithm:** L-BFGS optimization algorithm**.**

All NARX models used the Mean Squared Error (MSE) as the loss function. These choices were made to optimize the modeling performance of the networks.

A linear ARX model was also estimated for comparison with these NARX neural networks, in order to validate their use within the proposed approach. The ARX model is a classical linear model structure that predicts the current output as a linear combination of past outputs and inputs. The ARX model was configured as a MIMO system with the following parameters: number of past output terms set to $n_{a1} = n_{a2} = 6$; the number of past input terms set to $n_{b1} = n_{b2} = 6$; and the input–output delays set to $n_{k1} = n_{k2} = 1$ in terms of the number of samples.

The predicted outputs y_{1p} and y_{2p} are given by Equations (9) and (10) as linear combinations of past outputs and inputs, where $a_{1,i}$, $a_{2,i}$, $a_{3,i}$, $a_{4,i}$ and $b_{1,i}$, $b_{2,i}$, $b_{3,i}$, $b_{4,i}$ are constant model parameters estimated using the Ordinary Least Squares (OLS) method [73], which minimizes the Sum of Squared Errors (SSE). The parameters n_{a1} and n_{a2} represent the number of past outputs; n_{b1} and n_{b2} denote the number of past inputs; and n_{k1} and n_{k2} are the input–output delays. The variables u_1 and u_2 correspond to the control inputs for the fan and lamp, respectively.

$$y_{1p}(t) = -\sum_{i=1}^{n_{a1}} a_{1,i} y_{1p}(t-i) - \sum_{i=1}^{n_{a2}} a_{2,i} y_{2p}(t-i), + \sum_{i=n_{k1}}^{n_{b1}} b_{1,i} u_1(t-i) + \sum_{i=n_{k2}}^{n_{b2}} b_{2,i} u_2(t-i). \quad (9)$$

$$y_{2p}(t) = -\sum_{i=1}^{n_{a2}} a_{3,i} y_{2p}(t-i) - \sum_{i=1}^{n_{a1}} a_{4,i} y_{1p}(t-i), + \sum_{i=n_{k1}}^{n_{b1}} b_{3,i} u_1(t-i) + \sum_{i=n_{k2}}^{n_{b2}} b_{4,i} u_2(t-i). \quad (10)$$

The implementations were carried out in MATLAB (R2023a) using the Deep Learning Toolbox and System Identification Toolbox, and in Python 3.13.5 using the PyTorch library. The experiments were conducted on a computer with an Intel Core i7-1255U processor (1.70 GHz, 10 cores), 16 GB of RAM, running Windows 11.

6.2. Fault simulation. To acquire system data under faulty conditions, controlled fault scenarios were created. To this end, faults were introduced manually into the system. Actuator faults affect the system at the control level, whereas sensor faults manifest at the output level. Faults can manifest in various forms such as sudden steps, ramps, or other forms, depending on their nature.

We considered three different faulty states: one sensor and two actuator faults. The actuator faults were introduced as steps of different amplitudes: $f_{u2} = 60$ for the lamp fault, $f_{u1} = -75$ for the fan fault. The sensor fault, affecting the humidity measurement, varied within a range of $f_{y2} \in [3, 5]\%$. These faults were chosen to reflect plausible fault scenarios in the incubator's operating conditions.

6.3. Fault classification. To perform the isolation of the faults simulated in the system, different machine learning methods were used as fault classifiers, namely MLP, SVM, KNN, LDA, and XGBoost.

The MLP classifier has two inputs (corresponding to the residual vector dimension) and four outputs (representing the healthy and the three faulty states). Residual data were collected for each system state, resulting in four distinct datasets. The classifier was trained so that for each dataset, the output corresponding to that state was set to 1, while all others were set to 0.

In total, 9280 residual data samples were collected: 1600 for the healthy state, 2400 for actuator fault 1, 2640 for actuator fault 2, and 2640 for the sensor fault. These data were used to train the MLP classifier, with target assignments as described above.

The configuration of our neural classifier was determined as follows:

– **Number of hidden layers and neurons:** 1 hidden layer with 10 neurons.

– **Activation functions:** Sigmoid (hidden layer), Softmax (output layer).

– **Number of epochs:** 1000 epochs.

– **Optimization algorithm:** Bayesian regularization backpropagation algorithm.

– **Loss function:** Mean Squared Error.

These choices were made to optimize the network's classification performance.

The other methods were trained using the same two-dimensional residual vector as input, representing the system's deviations under various conditions, which is the same vector used for training the MLP classifier. However, they have a single output that assigns a specific label to each

distinct state: 1 for the healthy system, 2 for actuator fault 1, 3 for actuator fault 2 and 4 for the sensor fault.

For the SVM, the RBF kernel was selected for its capacity to capture nonlinear relationships within the data. This kernel is commonly used for classifying complex system states. To address multi-class classification problem, the Error-Correcting Output Code (ECOC) method was applied. This technique simplifies the multi-class problem by dividing it into several binary classification tasks, thereby enhancing the SVM's ability to differentiate among various fault conditions.

The applied XGBoost classifier is based on decision trees. It builds an ensemble of trees sequentially, with each new tree designed to correct the errors of the previous ones. The total number of trees was fixed at 100, with a maximum depth of 6 per tree. A learning rate of 0.3 was set to control the contribution of each tree to the final model. The loss function was multiclass logarithmic loss.

The KNN algorithm was employed. it determines the class of a sample by considering the K closest points in the feature space and assigning the most frequent class among them. The number of neighbors (K) was set to 5.

LDA was also applied. It operates by projecting the data onto a lower-dimensional space to maximize inter-class separation, thereby enabling fault classification.

The implementations were carried out in MATLAB R2023a (using the Deep Learning Toolbox and the Statistics and Machine Learning Toolbox) and in Python 3.13.5 (using the XGBoost library for the XGBoost classifier). The experiments were conducted on a computer with an Intel Core i7-1255U processor (1.70 GHz, 10 cores), 16 GB of RAM, running Windows 11.

7. Results and discussion. This section presents and analyzes the modeling and classification results.

7.1. Modeling results. To quantitatively assess the modeling performances of the considered models, three regression metrics were considered:

– **Mean Absolute Error:** This metric measures the average absolute difference between the predicted and actual values. It is calculated using the following formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{11}$$

where y_i denotes the actual incubator output at sample i ; $\hat{y}_i$ the corresponding predicted output from the model; and n the total number of

samples. A lower MAE indicates that predictions are, on average, closer to the actual values.

– **Root Mean Squared Error:** This metric computes the square root of the average of squared differences between predicted and actual values. It is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \quad (12)$$

RMSE penalizes larger errors more heavily, making it sensitive to significant deviations.

– **Mean Absolute Percentage Error:** This metric expresses the prediction error in relative terms, regardless of the scale of the data. It is calculated using the following formula:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \quad (13)$$

Table 1 presents the performance metrics of the models for the humidity and temperature outputs. A notable difference can be observed between the linear ARX model and the NARX neural networks. The ARX model exhibited higher error values for both outputs, with MAE values of 0.87 and 0.28 for humidity and temperature, respectively. This result indicates the limitation of linear modeling approaches, such as ARX, in capturing the behavior of systems with nonlinear dynamics.

Table 1. Model performance comparison

Model	Output	MAE	RMSE	MAPE
ARX	Humidity	0.87	1.04	1.25
	Temperature	0.28	0.34	0.85
MLP-NARX	Humidity	0.15	0.21	0.21
	Temperature	0.08	0.12	0.24
ELMAN-NARX	Humidity	0.29	0.37	0.42
	Temperature	0.09	0.12	0.29
LSTM-NARX	Humidity	0.24	0.30	0.33
	Temperature	0.10	0.13	0.32
GRU-NARX	Humidity	0.30	0.38	0.44
	Temperature	0.09	0.12	0.29

In contrast, the NARX models achieved low error metrics, demonstrating their effectiveness in modeling both outputs. Furthermore, the small differences between the RMSE and MAE values for these models indicate the absence of significant error peaks, as RMSE penalizes large errors more heavily. Performance differences among the NARX neural networks are more pronounced for humidity prediction, while temperature predictions remain relatively similar. The comparative study showed that the MLP-NARX model achieved the best results, with MAE values of 0.15 and 0.08 for humidity and temperature, respectively. The LSTM-NARX model ranked second, with MAE values of 0.24 for humidity and 0.10 for temperature. The Elman-NARX and GRU-NARX models demonstrated very similar, yet slightly lower, performance, with respective MAE values of 0.29 and 0.30 for humidity and an MAE of 0.09 for temperature. The strong performance of these different implementations of the NARX model highlights its suitability for modeling nonlinear dynamics.

Figure 5 illustrates the modeling results obtained with the selected MLP-NARX model by comparing the actual and predicted outputs of the incubator. The figure shows the last 1200 samples of the training dataset and 1680 samples from the test set, with actual and predicted values plotted for each output. The close alignment between the curves visualizes the low prediction error.

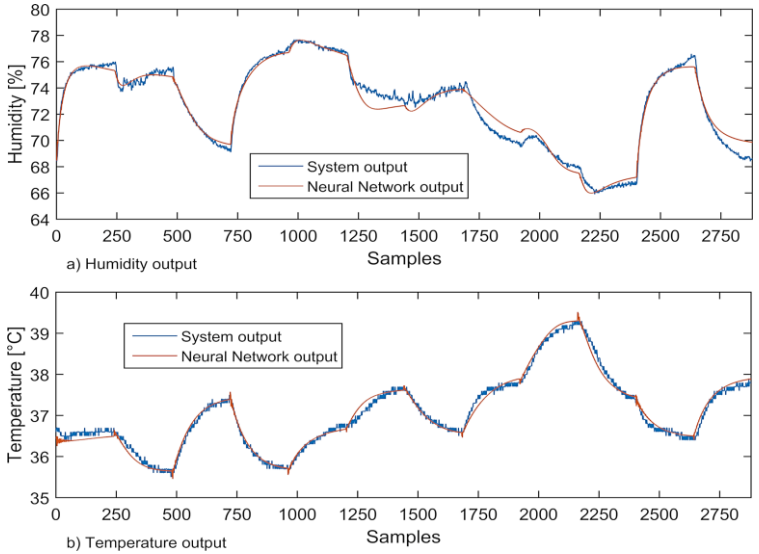


Fig. 5. MLP-NARX Modeling results: comparison between actual and predicted outputs: a) Humidity output, b) Temperature output

7.2. Classification results. The classifiers were evaluated using a structured dataset comprising a total of 2160 samples. The first 250 samples were from the fault-free condition, followed by 250 samples representing the sensor fault condition. An additional 220 healthy-state samples were then included. Subsequently, the dataset included 720 samples for actuator fault 2 and 720 samples for actuator fault 1. Table 2 lists the data samples with their corresponding system states.

Table 2. Data samples with their corresponding system states

Data samples	System state
1-250	Healthy system
251-500	Sensor fault
501-720	Healthy system
721-1440	Actuator fault 2
1441-2160	Actuator fault 1

The fault isolation was achieved using five machine learning algorithms: MLP, SVM, KNN, LDA, and XGBoost. Their effectiveness for the fault classification task was evaluated through a comparative study based on the following performance:

– **Accuracy:** It measures the overall proportion of correct predictions, providing a global indicator of the classifier's performance. It is calculated using the following formula:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}. \quad (14)$$

– **Precision:** It measures the proportion of true positive predictions cases among all positive predictions. It is calculated using the following formula:

$$Precision = \frac{TP}{TP + FP}, \quad (15)$$

where, for a given class, True Positives (TP) denote samples of the class correctly identified; True Negatives (TN) are samples from other classes correctly rejected; False Positives (FP) correspond to samples incorrectly assigned to the class; and False Negatives (FN) are samples of the class that the model failed to recognize. High precision indicates a low rate of false positive errors.

– **Recall:** It measures the proportion of actual positive cases that are correctly identified. It is calculated using the following formula:

$$Recall = \frac{TP}{TP + FN}. \quad (16)$$

High recall indicates that the model successfully captures most of the relevant positive cases.

– **F_1-score:** It is the harmonic mean of precision and recall. It is calculated using the following formula:

$$F_1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (17)$$

It provides a balanced measure between precision and recall, making it particularly useful when both false positives and false negatives are critical, as is the case in our FDI task.

Precision, recall, and F_1-score were calculated for each class and then averaged using the macro-averaging method, which assigns equal weight to all classes.

Table 3 presents the performance metrics of the different ML algorithms for the fault classification task. The comparative study of the different classifiers showed that the MLP classifier achieved the highest results across all metrics, with an accuracy of 95.8%, precision of 96.5%, recall of 96.8%, and F_1-score of 96.6%. These results indicate that the MLP is both highly accurate in its predictions and balanced in terms of precision and recall, making it effective at minimizing both false positives and false negatives, which is crucial for our FDI task.

Table 3. Performance comparison of machine learning classifiers

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)
MLP	95.8	96.5	96.8	96.6
LDA	88.8	86.8	89.4	86.1
XGBOOST	95.4	96.1	96.5	96.2
SVM	90.8	93.4	93.1	92.7
KNN	86.9	91.7	90.2	89.6

The XGBoost classifier was the second-best performer, with an accuracy of 95.4% and an F_1-score of 96.2%. It demonstrated competitive

performance, confirming its suitability for classification tasks, particularly in handling nonlinear relationships.

The SVM also achieved strong results, with an F_1-score of 92.7%. However, it underperformed relative to the top two methods, indicating a lower ability to capture complex data patterns.

LDA and KNN achieved comparatively lower performance. LDA assumes linear class boundaries, which limits its effectiveness on complex nonlinear data, while KNN struggles to capture complex and global relationships within the dataset. They exhibited contrasting performance patterns. LDA tended to favor the majority class, resulting in higher accuracy but lower precision, recall, and F_1-score. The low F_1-score for LDA is a direct consequence of the imbalance between its precision and recall. Conversely, KNN produced a more balanced distribution of predictions across classes, which boosted its precision, recall, and F_1-score but resulted in a drop in overall accuracy.

In conclusion, the results demonstrate that the MLP classifier, owing to its strong ability to capture complex nonlinear relationships, achieved the best fault classification performance, underscoring its high potential for practical FDI applications.

8. Conclusion. In this paper, an intelligent FDI methodology was proposed, using an incubator system as a case study. The design and implementation of the incubator were presented. The complex nonlinear system dynamics were modeled using a NARX neural network in a parallel structure. A comparative study was conducted among different neural networks implementing this model—MLP, LSTM, GRU, and Elman—to identify the best-performing architecture. A linear ARX model was also estimated for comparison with these NARX neural networks to validate their use within the proposed approach. The obtained residuals were then used for FDI using five classification methods: MLP, XGBoost, SVM, KNN, and LDA, whose performance was compared. The results demonstrated the effectiveness of the proposed FDI methodology. The NARX models exhibited strong modeling performance, outperforming the ARX model, with the MLP-NARX attaining the best results. The comparative study of classifiers showed that the MLP classifier achieved the best performance across all evaluation metrics, further highlighting its potential for FDI system.

References

1. Mujcic E., Drakulic U. Design and Implementation of Fuzzy Control System for Egg Incubator based on IoT Technology. IOP Conference Series: Materials Science and Engineering. 2021. vol. 1208. no. 1. DOI: 10.1088/1757-899X/1208/1/012038.

2. Mitchaothai J., Lertpatarakomol R., Trairatapiwan T., Lukkananukool A. Influence of Incubation Temperature and Relative Humidity on the Egg Hatchability Pattern of Two-Spotted (Gryllus bimaculatus) and House (Acheta domesticus) Crickets. Animals. 2024. vol. 14. no. 15. DOI: 10.3390/ani14152176.
3. Noiva R.M., Menezes A.C., Peleteiro M.C. Influence of Temperature and Humidity Manipulation on Chicken Embryonic Development. BMC Veterinary Research. 2014. vol. 10. pp. 1–10. DOI: 10.1186/s12917-014-0234-3.
4. Yakimov V.L., Maltsev G.N. Hybrid Network Structures and Their Use in Diagnosing Complex Technical Systems. Informatics and Automation. 2022. vol. 21. no. 1. pp. 126–160. DOI: 10.15622/ia.2022.21.5.
5. Isermann R. Model-Based Fault-Detection and Diagnosis – Status and Applications. Annual Reviews in Control. 2005. vol. 29. no. 1. pp. 71–85. DOI: 10.1016/j.arcontrol.2004.12.002.
6. Patton R.J. Fault Detection and Diagnosis in Aerospace Systems Using Analytical Redundancy. Computer Control Engineering Journal. 1991. vol. 2. no. 3. pp. 127–136. DOI: 10.1049/cce:19910031.
7. Bernardi E., Adam E.J. Observer-Based Fault Detection and Diagnosis Strategy for Industrial Processes. Journal of the Franklin Institute. 2020. vol. 357. no. 14. pp. 10054–10081. DOI: 10.1016/j.jfranklin.2020.07.046.
8. Mo J., Qin D., Liu Y. Unknown Input Observer-Based Fault Diagnosis of Speed Sensors in Dual Clutch Transmission. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering. 2022. vol. 237. no. 7. pp. 1710–1720. DOI: 10.1177/09544070221095286.
9. Khentout N., Salhi H., Magrotti G., Merrouche D. Fault Monitoring and Accommodation of the Heat Exchanger Parameters of Triga-Mark II Nuclear Research Reactor Using Model-Based Analytical Redundancy. Progress in Nuclear Energy. 2018. vol. 109. pp. 97–112. DOI: 10.1016/j.pnucene.2018.02.019.
10. Jihani N., Kabbaj M.N., Benbrahim M. Kalman Filter Based Sensor Fault Detection in Wireless Sensor Network for Smart Irrigation. Results in Engineering. 2023. vol. 20. DOI: 10.1016/j.rineng.2023.101395.
11. Patan K. Artificial Neural Networks for the Modeling and Fault Diagnosis of Technical Processes. Springer, 2008. 206 p.. DOI: 10.1007/978-3-540-79872-9.
12. Patan K., Korbicz J. Artificial Neural Networks in Fault Diagnosis. Berlin, Heidelberg: Springer Berlin Heidelberg. 2004. pp. 333–379. DOI: 10.1007/978-3-642-18615-8_9.
13. Yu D.L., Hamad A., Gomm J.B., Sangha M.S. Dynamic Fault Detection and Isolation for Automotive Engine Air Path by Independent Neural Network Model. International Journal of Engine Research. 2014. vol. 15. no. 1. pp. 87–100. DOI: 10.1177/1468087412461267.
14. Mohd Amiruddin A.A., Zabiri H., Taqvi S.A.A., Tufa L.D. Neural Network Applications in Fault Diagnosis and Detection: An Overview of Implementations in Engineering-Related Systems. Neural Computing and Applications. 2020. vol. 32. no. 2. pp. 447–472. DOI: 10.1007/s00521-018-3911-5.
15. Atmane M.A., Boudebouda O., Zatla H., Nouar S.F., Tolbi B., Djeriri Y., Bouhamama M. Fault Diagnosis: An Integrated Methodology Based on Neural Networks. Proceedings of 3rd International Conference on Advanced Electrical Engineering (ICAEE'2024). IEEE, 2024. pp. 1–6. DOI: 10.1109/ICAEE61760.2024.10783413.
16. Nasser R., Azar A.T., Humaidi A.J., Al-Mhdawi A.K., Ibraheem I.K. Intelligent Fault Detection and Identification Approach for Analog Electronic Circuits Based on Fuzzy Logic Classifier. Electronics. 2021. vol. 10. no. 23. DOI: 10.3390/electronics10232888.

17. Yang L., Gao L., Luo X., Hao Y., Zhang Z., Jin Y., Zhang J. Improved Method for Fault Diagnosis of Oil-Immersed Transformers Based on Simulation Test Platform. IEEE Transactions on Dielectrics and Electrical Insulation. 2025. vol. 32. no. 1. pp. 571–580. DOI: 10.1109/TDEI.2024.3418388.
18. Venkata P., Pandya V., Vala K., Sant A.V. Support Vector Machine for Fast Fault Detection and Classification in Modern Power Systems Using Quarter Cycle Data. Energy Reports. 2022. vol. 8. pp. 92–98. DOI: 10.1016/j.egyr.2022.10.279.
19. Tuerxun W., Chang X., Hongyu G., Zhijie J., Huajian Z. Fault Diagnosis of Wind Turbines Based on a Support Vector Machine Optimized by the Sparrow Search Algorithm. IEEE Access. 2021. vol. 9. pp. 69307–69315. DOI: 10.1109/ACCESS.2021.3075547.
20. Cho S., Choi M., Gao Z., Moan T. Fault Detection and Diagnosis of a Blade Pitch System in a Floating Wind Turbine Based on Kalman Filters and Artificial Neural Networks. Renewable Energy. 2021. vol. 169. pp. 1–13. DOI: 10.1016/j.renene.2020.12.116.
21. Wang Y., Wen C., Wu X. Fault Detection and Isolation of Floating Wind Turbine Pitch System Based on Kalman Filter and Multi-Attention 1DCNN. Systems Science & Control Engineering. 2024. vol. 12. no. 1. DOI: 10.1080/21642583.2024.2362169.
22. Yu D.L., Gomm J.B., Williams D. Sensor Fault Diagnosis in a Chemical Process via RBF Neural Networks. Control Engineering Practice. 1999. vol. 7. no. 1. pp. 49–55. DOI: 10.1016/S0967-0661(98)00167-1.
23. Liu X., He H. Fault Diagnosis for TE Process Using RBF Neural Network. IEEE Access. 2021. vol. 9. pp. 118453–118460. DOI: 10.1109/ACCESS.2021.3107360.
24. Rahimilarki R., Gao Z., Jin N., Zhang A. Convolutional Neural Network Fault Classification Based on Time-Series Analysis for Benchmark Wind Turbine Machine. Renewable Energy. 2022. vol. 185. pp. 916–931. DOI: 10.1016/j.renene.2021.12.056.
25. Choudhary A., Mian T., Fatima S. Convolutional Neural Network Based Bearing Fault Diagnosis of Rotating Machine Using Thermal Images. Measurement. 2021. vol. 176. DOI: 10.1016/j.measurement.2021.109196.
26. Zhang Y., Zhou T., Huang X., Cao L., Zhou Q. Fault Diagnosis of Rotating Machinery Based on Recurrent Neural Networks. Measurement. 2021. vol. 171. DOI: 10.1016/j.measurement.2020.108774.
27. Zhu J., Jiang Q., Shen Y., Qian C., Xu F., Zhu Q. Application of Recurrent Neural Network to Mechanical Fault Diagnosis: A Review. Journal of Mechanical Science and Technology. 2022. vol. 36. no. 2. pp. 527–542. DOI: 10.1007/s12206-022-0102-1.
28. An Y., Sun X., Ren B., Li H., Zhang M. A Data-driven Method for IGBT Open-circuit Fault Diagnosis for the Modular Multilevel Converter based on a modified Elman Neural Network. Energy Reports. 2022. vol. 8. pp. 80–88. DOI: 10.1016/j.egyr.2022.08.024.
29. Nguyen V.-Q., Vu M.-H., Pham V.-T., Tran T.-T. A Deep Learning Approach Based on MLP-Mixer Models for Bearing Fault Diagnosis. Proceedings of International Conference on System Science and Engineering (ICSSE'2023). IEEE, 2023. pp. 16–21. DOI: 10.1109/ICSSE58758.2023.10227218.
30. Khoualdia T., Lakehal A., Chelli Z., Khoualdia K., Nessaib K. Optimized Multi Layer Perceptron Artificial Neural Network Based Fault Diagnosis of Induction Motor Using Vibration Signals. Diagnostyka. 2021. vol. 22. no. 1. pp. 65–74. DOI: 10.29354/diag/133091.
31. Naimi A., Deng J., Shimjith S.R., Arul A.J. Fault Detection and Isolation of a Pressurized Water Reactor based on Neural Network and K-Nearest Neighbor. IEEE Access. 2022. vol. 10. pp. 17113–17121. DOI: 10.1109/ACCESS.2022.3149772.

32. Ma Y., Liu H., Zhu Y., Wang F., Luo Z. The NARX Model-Based System Identification on Nonlinear, Rotor-Bearing Systems. Applied Sciences. 2017. vol. 7. no. 9. pp. 911. DOI: 10.3390/app7090911.
33. Choe H.O., Lee M.H. Artificial Intelligence-based Fault Diagnosis and Prediction for Smart Farm Information and Communication Technology Equipment. Agriculture. 2023. vol. 13. no. 11. DOI: 10.3390/agriculture13112124.
34. Taqvi S.A., Tufa L.D., Zabiri H., Maulud A.S., Uddin F. Fault Detection in Distillation Column Using NARX Neural Network. Neural Computing and Applications. 2020. vol. 32. no. 8. pp. 3503–3519. DOI: 10.1007/s00521-018-3658-z.
35. Mustafaraj G., Lowry G., Chen J. Prediction of Room Temperature and Relative Humidity by Autoregressive Linear and Nonlinear Neural Network Models for an Open Office. Energy and Buildings. 2011. vol. 43. no. 6. pp. 1452–1460. DOI: 10.1016/j.enbuild.2011.02.007.
36. Sani M.G., Abdul Wahab N., Sam Y.M., Samsudin S.I., Jamaludin I.W. Comparison of NARX Neural Network and Classical Modelling Approaches. Applied mechanics and Materials. 2014. vol. 554. pp. 360–365. DOI: 10.4028/www.scientific.net/AMM.554.360.
37. Lin Y.P. ARX and NARX Modeling and Control of a Continuous UV/H2O2 Photoreactor for the Aqueous PVA Degradation. Master's Thesis. Toronto Metropolitan University, 2024. 171 p. DOI: 10.32920/25761546.
38. Ling T.G., Rahmat M.F., Husain A.R. A Comparative Study of Linear ARX and Nonlinear ANFIS Modeling of an Electro-hydraulic Actuator System. Jurnal Teknologi (Sciences & Engineering). 2014. vol. 67. no. 5. pp. 1–8. DOI: 10.11113/jt.v67.2833.
39. Renteria-Mena J.B., Plaza D., Giraldo E. Multivariable NARX based Neural Networks Models for Short-term Water Level Forecasting. Engineering Proceedings. 2023. vol. 39. no. 1. DOI: 10.3390/engproc2023039060.
40. Delcroix B., Ny J.L., Bernier M., Azam M., Qu B., Venne J.S. Autoregressive Neural Networks with Exogenous Variables for Indoor Temperature Prediction in Buildings. Building simulation. 2021. vol. 14. no. 1. pp. 165–178. DOI: 10.1007/s12273-019-0597-2.
41. Ramirez C., Acuna G. Forecasting Cash Demand in ATM using Neural Networks and Least Square Support Vector Machine. Proceedings of 16th Iberoamerican Congress on Pattern Recognition (CIARP). Springer Berlin Heidelberg, 2011. pp. 515–522. DOI: 10.1007/978-3-642-25085-9_61.
42. Shao Y., Zhao L. NARX-GA-Elman Method for Mach Number Prediction of Wind Tunnel Flow Field. Instrumentation. 2023. vol. 10. no. 4. DOI: 10.15878/j.cnki.instrumentation.2023.04.004.
43. Nikentari N., Wei H.L. Tide Level Prediction Using NARX-based Recurrent Neural Networks. Proceeding of 27th International Conference on Automation and Computing (ICAC). IEEE, 2022. pp. 1–6. DOI: 10.1109/ICAC55051.2022.9911163.
44. Freitas V.C.G.D., Araujo V.G.D., Crisostomo D.C.D.C., Lima G.F.D., Neto A.D.D., Salazar A.O. Velocity Prediction of a Pipeline Inspection Gauge (PIG) with Machine Learning. Sensors. 2022. vol. 22. no. 23. DOI: 10.3390/s22239162.
45. Ghate V.N., Dudul S.V. Optimal MLP Neural Network Classifier for Fault Detection of Three Phase Induction Motor. Expert Systems with Applications. 2010. vol. 37. no. 4. pp. 3468–3481. DOI: 10.1016/j.eswa.2009.10.041.
46. Teler K., Skowron M., Orłowska-Kowalska T. Implementation of MLP-Based Classifier of Current Sensor Faults in Vector-Controlled Induction Motor Drive. IEEE Transactions on Industrial Informatics. 2024. vol. 20. no. 4. pp. 5702–5713. DOI: 10.1109/TII.2023.3336348.

47. Lin W., Yang C., Lin J., Tsay M.A. Fault Classification Method by RBF Neural Network with OLS Learning Procedure. IEEE Power Engineering Review. 2001. vol. 21. no. 8. DOI: 10.1109/MPER.2001.4311561.
48. Yang P., Wang T., Yang H., Meng C., Zhang H., Cheng L. The Performance of Electronic Current Transformer Fault Diagnosis Model: Using an Improved Whale Optimization Algorithm and RBF Neural Network. Electronics. 2023. vol. 12. no. 4. DOI: 10.3390/electronics12041066.
49. Jeong K., Choi S.B., Choi H. Sensor Fault Detection and Isolation Using a Support Vector Machine for Vehicle Suspension Systems. IEEE Transactions on Vehicular Technology. 2020. vol. 69. no. 4. pp. 3852–3863. DOI: 10.1109/TVT.2020.2977353.
50. Ramdani O., Beddek K., Haddouche R., Zerrougui M., Chouider N. SVM-Based Approach Fault Detection for PMSG-Wind Energy Conversion System. Journal of Engineering Research. 2025. vol. 13. no. 4. pp. 3378–3393. DOI: 10.1016/j.jer.2025.01.001.
51. Zhou N., Shao Q., Zhou J., Changjie H. Fault Classification of Proton Exchange Membrane Fuel Cells for Vehicles based on XGBoost. Proceedings of 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2021. pp. 1054–1058. DOI: 10.1109/ICBAIE52039.2021.9389943.
52. Hasan S., Toufikuzzaman M. Fault Occurrence Detection and Classification of Fault Type in Electrical Power Transmission Line with Machine Learning Algorithms. International Journal on Electrical Engineering and Informatics. 2022. vol. 14. no. 3. DOI: 10.15676/ijeei.2022.14.3.9.
53. Mehta A., Goyal D., Choudhary A., Pabla B.S., Belghith S. Machine Learning-Based Fault Diagnosis of Self-Aligning Bearings for Rotating Machinery Using Infrared Thermography. Mathematical Problems in Engineering. 2021. vol. 2021. no. 1. DOI: 10.1155/2021/9947300.
54. Shaikh F.A., Kamboh M.Z., Alvi B.A., Khan S., Khan F.M. Condition-Based Health Monitoring of Electrical Machines Using DWT and LDA Classifier. Sir Syed University Research Journal of Engineering and Technology. 2022. vol. 12. no. 2. pp. 95–100. DOI: 10.33317/ssurj.513.
55. Altayef E., Anayi F., Packianather M., Benmahamed Y., Kherif O. Detection and Classification of Lamination Faults in A 15 kVA Three-Phase Transformer Core using SVM, KNN and DT Algorithms. IEEE Access. 2022. vol. 10. pp. 50925–50932. DOI: 10.1109/ACCESS.2022.3174359.
56. Shinde P.V., Desavale R.G., Jadhav P.M., Sawant S.H. A Multi Fault Classification in a Rotor-Bearing System using Machine Learning Approach. Journal of the Brazilian Society of Mechanical Sciences and Engineering. 2023. vol. 45. no. 2. DOI: 10.1007/s40430-023-04015-1.
57. Rakhmawati R., Murdianto F.D., Luthfi A., Rahman A.Y. Thermal Optimization on Incubator using Fuzzy Inference System based IoT. Proceedings of International Conference of Artificial Intelligence and Information Technology (ICAIIT). IEEE, 2019. pp. 464–468. DOI: 10.1109/ICAIIT.2019.8834530.
58. Febriani A., Wahyuni R., Mardeni M., Irawan Y., Melyanti R. Improved Hybrid Machine and Deep Learning Model for Optimization of Smart Egg Incubator. Journal of Applied Data Sciences. 2024. vol. 5. no. 3. pp. 1052–1068. DOI: 10.47738/jads.v5i3.304.
59. Easwaran A., Arvindan P., Dhanyasree E., Surya R., Selvakumar S. Internet of Things Enabled Smart Animal Farm Prototype. Journal of Physics: Conference Series. 2021. vol. 2070. no. 1. DOI: 10.1088/1742-6596/2070/1/012115.
60. Guzman-Zabala J.V., Castro-Martin A.P. Smart and Semi-industrial Egg Incubator with Remote Monitoring Using LoRa Technology. Proceedings of International Conference on

Computer Science, Electronics and Industrial Engineering (CSEI). Springer Nature Switzerland, 2024. pp. 522–540. DOI: 10.1007/978-3-031-70981-4_35.
61. Roshanghiyasi H., Ahmad A.H., Hosseinpour S., Jafari A., Mousazadeh H., Asadollahzadeh A. Monitoring and Controlling the Chicken Incubation Process Using the Internet of Things System. Journal of Agricultural Mechanization. 2022. vol. 7. no. 1. pp. 73–77. DOI: 10.22034/jam.2022.15710.
62. Radhakrishnan K., Jose N., Sanjay S.G., Cherian T., Vishnu K.R. Design and Implementation of a Fully Automated Egg Incubator. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2014. vol. 3. no. 2. pp. 7686–7690.
63. Uzoigwe L.O., Ekezie J.C. Egg Incubator Control System with Short Message Service (sms) Fault Analysis Alert. Journal of Agriculture and Food Sciences. 2013. vol. 11. no. 2. pp. 45–68. DOI: 10.4314/jafs.v11i2.5.
64. Stull R. Practical Meteorology: An Algebra-based Survey of Atmospheric Science. University of British Columbia, 2017. 940 p.
65. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Representations by Back-Propagating Errors. Nature. 1986. vol. 323. no. 6088. pp. 533–536. DOI: 10.1038/323533a0.
66. Elman J.L. Finding Structure in Time. Cognitive science. 1990. vol. 14. no. 2. pp. 179–211. DOI: 10.1207/s15516709cog1402_1.
67. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural computation. 1997. vol. 9. no. 8. pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
68. Cho K., Van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014. pp. 1724–1734. DOI: 10.3115/v1/D14-1179.
69. MacKay D.J.C. Bayesian Interpolation. Neural Computation. 1992. vol. 4. no. 3. pp. 415–447. DOI: 10.1162/neco.1992.4.3.415.
70. Londhe A.S., Ingale A.S., Khadse C.B. Bayesian Regularization Neural Network-based Fault Detection System in HVDC Transmission System. Proceedings of International Conference on Smart Technologies for Energy, Environment, and Sustainable Development (ICSTEESD). Springer Nature, 2022. pp. 601–607. DOI: 10.1007/978-981-16-6875-3_48.
71. Hassan M.S., Kamal K., Ratlamwala T.A.H. Fault Classification of Power Plants using Artificial Neural Network. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. 2022. vol. 44. no. 3. pp. 7665–7680. DOI: 10.1080/15567036.2022.2113936.
72. Liu D.C., Nocedal J. On the Limited Memory BFGS Method for Large Scale Optimization. Mathematical Programming. 1989. vol. 45. no. 1. pp. 503–528. DOI: 10.1007/BF01589116
73. Ljung L. System Identification: Theory for the User. Prentice Hall, 1999. pp. 640.

Atmane Mohamed — Ph.D. student, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: fault detection and isolation, artificial intelligence, automation. The number of publications — 1. mohamed.atmane@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Zatla Hicham — Ph.D., Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: automatic control, artificial intelligence, human-machine systems. The number of publications — 8.

hicham.zatla@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Tolbi Bilal — Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: artificial intelligence, hybrid systems, systems control. The number of publications — 15. bilal.tolbi@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Nouar Souad — Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: physical system modeling, numerical simulation. The number of publications — 2. souadfadila.nouar@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Bouhamama Mohamed — Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: electrical engineering, environmental technologies. The number of publications — 14. mohammed.bouhamama@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Acknowledgements. This research is supported by Directorate General for Scientific Research and Technological Development (DGRSDT).

DOI 10.15622/ia.25.2.6

М.А. АТМАНЕ, Х. ЗАТЛА, Б. ТОЛБИ, С.Ф. НУАР, М. БУХАМАМА

ИНТЕЛЛЕКТУАЛЬНОЕ ОБНАРУЖЕНИЕ И ИЗОЛЯЦИЯ НЕИСПРАВНОСТЕЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ NARX

Атмане М.А., Затла Х., Толби Б., Нуар С.Ф., Бухамама М. **Интеллектуальное обнаружение и изоляция неисправностей на основе нейронных сетей NARX.**

Аннотация. Интеллектуальные системы стали неотъемлемым компонентом современных технологических ландшафтов. Вопрос их надежности является крайне важным, поскольку возникновение неисправностей способно катастрофически сказаться на функционировании и итоговой эффективности системы, вследствие чего обнаружение и изоляция неисправностей (FDI) становится критически значимой задачей. Реализация этой задачи осложняется присущей таким системам сложной нелинейной динамикой. В данной работе в рамках решения указанной проблемы предлагается методология интеллектуального обнаружения и изоляции неисправностей; в качестве репрезентативного примера рассматривается система инкубатора. Предлагаемый метод использует нелинейную авторегрессионную экзогенную (NARX) нейронную сеть в параллельной структуре для моделирования сложной нелинейной динамики системы. Были сравнены различные реализации модели NARX, основанные на многослойном перцептроне (MLP), сети долгой краткосрочной памяти (LSTM), вентильном рекуррентном блоке (GRU) и сети Элмана, для оценки их эффективности моделирования и определения лучшей модели. Полученные расхождения между прогнозами модели и фактическими значениями называются остаточными величинами. Для классификации неисправностей было проведено сравнительное исследование пяти методов машинного обучения (ML): многослойного перцептрона, экстремального градиентного бустинга (XGBoost), метода опорных векторов (SVM), метода k-ближайших соседей (KNN) и линейного дискриминантного анализа (LDA). Данные методы анализируют остаточные величины, чтобы идентифицировать конкретную неисправность из заранее определенного множества возможных, включая неисправности исполнительных механизмов и датчиков. Полученные результаты демонстрируют эффективность интеллектуальной методологии FDI, представленной в данной работе. Модели NARX продемонстрировали высокую эффективность моделирования, причем гибридная модель MLP-NARX показала наилучшие результаты, превзойдя остальные архитектуры. Сравнительный анализ классификаторов выявил различия в эффективности пяти рассматриваемых методов, при этом классификатор на основе многослойного перцептрона достиг наивысших показателей по всем оценочным метрикам. Это свидетельствует о его пригодности для практического применения в задачах диагностики неисправностей, что обусловлено его высокой способностью улавливать сложные нелинейные взаимосвязи в данных.

Ключевые слова: обнаружение и изоляция неисправностей, инкубатор, многослойный перцептрон, моделирование NARX, генерация и оценка остаточных величин, рекуррентные нейронные сети, классификаторы машинного обучения, интеллектуальные системы.

Литература

1. Mujcic E., Drakulic U. Design and Implementation of Fuzzy Control System for Egg Incubator based on IoT Technology. IOP Conference Series: Materials Science and Engineering. 2021. vol. 1208. no. 1. DOI: 10.1088/1757-899X/1208/1/012038.

2. Mitchaothai J., Lertpatarakomol R., Trairatapiwan T., Lukkananukool A. Influence of Incubation Temperature and Relative Humidity on the Egg Hatchability Pattern of Two-Spotted (Gryllus bimaculatus) and House (Acheta domesticus) Crickets. Animals. 2024. vol. 14. no. 15. DOI: 10.3390/ani14152176.
3. Noiva R.M., Menezes A.C., Peleteiro M.C. Influence of Temperature and Humidity Manipulation on Chicken Embryonic Development. BMC Veterinary Research. 2014. vol. 10. pp. 1–10. DOI: 10.1186/s12917-014-0234-3.
4. Yakimov V.L., Maltsev G.N. Hybrid Network Structures and Their Use in Diagnosing Complex Technical Systems. Informatics and Automation. 2022. vol. 21. no. 1. pp. 126–160. DOI: 10.15622/ia.2022.21.5.
5. Isermann R. Model-Based Fault-Detection and Diagnosis – Status and Applications. Annual Reviews in Control. 2005. vol. 29. no. 1. pp. 71–85. DOI: 10.1016/j.arcontrol.2004.12.002.
6. Patton R.J. Fault Detection and Diagnosis in Aerospace Systems Using Analytical Redundancy. Computer Control Engineering Journal. 1991. vol. 2. no. 3. pp. 127–136. DOI: 10.1049/cce:19910031.
7. Bernardi E., Adam E.J. Observer-Based Fault Detection and Diagnosis Strategy for Industrial Processes. Journal of the Franklin Institute. 2020. vol. 357. no. 14. pp. 10054–10081. DOI: 10.1016/j.jfranklin.2020.07.046.
8. Mo J., Qin D., Liu Y. Unknown Input Observer-Based Fault Diagnosis of Speed Sensors in Dual Clutch Transmission. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering. 2022. vol. 237. no. 7. pp. 1710–1720. DOI: 10.1177/09544070221095286.
9. Khentout N., Salhi H., Magrotti G., Merrouche D. Fault Monitoring and Accommodation of the Heat Exchanger Parameters of Triga-Mark II Nuclear Research Reactor Using Model-Based Analytical Redundancy. Progress in Nuclear Energy. 2018. vol. 109. pp. 97–112. DOI: 10.1016/j.pnucene.2018.02.019.
10. Jihani N., Kabbaj M.N., Benbrahim M. Kalman Filter Based Sensor Fault Detection in Wireless Sensor Network for Smart Irrigation. Results in Engineering. 2023. vol. 20. DOI: 10.1016/j.rineng.2023.101395.
11. Patan K. Artificial Neural Networks for the Modeling and Fault Diagnosis of Technical Processes. Springer, 2008. 206 p.. DOI: 10.1007/978-3-540-79872-9.
12. Patan K., Korbicz J. Artificial Neural Networks in Fault Diagnosis. Berlin, Heidelberg: Springer Berlin Heidelberg. 2004. pp. 333–379. DOI: 10.1007/978-3-642-18615-8_9.
13. Yu D.L., Hamad A., Gomm J.B., Sangha M.S. Dynamic Fault Detection and Isolation for Automotive Engine Air Path by Independent Neural Network Model. International Journal of Engine Research. 2014. vol. 15. no. 1. pp. 87–100. DOI: 10.1177/1468087412461267.
14. Mohd Amiruddin A.A., Zabiri H., Taqvi S.A.A., Tufa L.D. Neural Network Applications in Fault Diagnosis and Detection: An Overview of Implementations in Engineering-Related Systems. Neural Computing and Applications. 2020. vol. 32. no. 2. pp. 447–472. DOI: 10.1007/s00521-018-3911-5.
15. Atmane M.A., Boudebouda O., Zatla H., Nouar S.F., Tolbi B., Djeriri Y., Bouhamama M. Fault Diagnosis: An Integrated Methodology Based on Neural Networks. Proceedings of 3rd International Conference on Advanced Electrical Engineering (ICAEE'2024). IEEE, 2024. pp. 1–6. DOI: 10.1109/ICAEE61760.2024.10783413.
16. Nasser R., Azar A.T., Humaidi A.J., Al-Mhdawi A.K., Ibraheem I.K. Intelligent Fault Detection and Identification Approach for Analog Electronic Circuits Based on Fuzzy Logic Classifier. Electronics. 2021. vol. 10. no. 23. DOI: 10.3390/electronics10232888.
17. Yang L., Gao L., Luo X., Hao Y., Zhang Z., Jin Y., Zhang J. Improved Method for Fault Diagnosis of Oil-Immersed Transformers Based on Simulation Test Platform.

IEEE Transactions on Dielectrics and Electrical Insulation. 2025. vol. 32. no. 1. pp. 571–580. DOI: 10.1109/TDEI.2024.3418388.
18. Venkata P., Pandya V., Vala K., Sant A.V. Support Vector Machine for Fast Fault Detection and Classification in Modern Power Systems Using Quarter Cycle Data. Energy Reports. 2022. vol. 8. pp. 92–98. DOI: 10.1016/j.egyr.2022.10.279.
19. Tuerxun W., Chang X., Hongyu G., Zhijie J., Huajian Z. Fault Diagnosis of Wind Turbines Based on a Support Vector Machine Optimized by the Sparrow Search Algorithm. IEEE Access. 2021. vol. 9. pp. 69307–69315. DOI: 10.1109/ACCESS.2021.3075547.
20. Cho S., Choi M., Gao Z., Moan T. Fault Detection and Diagnosis of a Blade Pitch System in a Floating Wind Turbine Based on Kalman Filters and Artificial Neural Networks. Renewable Energy. 2021. vol. 169. pp. 1–13. DOI: 10.1016/j.renene.2020.12.116.
21. Wang Y., Wen C., Wu X. Fault Detection and Isolation of Floating Wind Turbine Pitch System Based on Kalman Filter and Multi-Attention 1DCNN. Systems Science & Control Engineering. 2024. vol. 12. no. 1. DOI: 10.1080/21642583.2024.2362169.
22. Yu D.L., Gomm J.B., Williams D. Sensor Fault Diagnosis in a Chemical Process via RBF Neural Networks. Control Engineering Practice. 1999. vol. 7. no. 1. pp. 49–55. DOI: 10.1016/S0967-0661(98)00167-1.
23. Liu X., He H. Fault Diagnosis for TE Process Using RBF Neural Network. IEEE Access. 2021. vol. 9. pp. 118453–118460. DOI: 10.1109/ACCESS.2021.3107360.
24. Rahimilarki R., Gao Z., Jin N., Zhang A. Convolutional Neural Network Fault Classification Based on Time-Series Analysis for Benchmark Wind Turbine Machine. Renewable Energy. 2022. vol. 185. pp. 916–931. DOI: 10.1016/j.renene.2021.12.056.
25. Choudhary A., Mian T., Fatima S. Convolutional Neural Network Based Bearing Fault Diagnosis of Rotating Machine Using Thermal Images. Measurement. 2021. vol. 176. DOI: 10.1016/j.measurement.2021.109196.
26. Zhang Y., Zhou T., Huang X., Cao L., Zhou Q. Fault Diagnosis of Rotating Machinery Based on Recurrent Neural Networks. Measurement. 2021. vol. 171. DOI: 10.1016/j.measurement.2020.108774.
27. Zhu J., Jiang Q., Shen Y., Qian C., Xu F., Zhu Q. Application of Recurrent Neural Network to Mechanical Fault Diagnosis: A Review. Journal of Mechanical Science and Technology. 2022. vol. 36. no. 2. pp. 527–542. DOI: 10.1007/s12206-022-0102-1.
28. An Y., Sun X., Ren B., Li H., Zhang M. A Data-driven Method for IGBT Open-circuit Fault Diagnosis for the Modular Multilevel Converter based on a modified Elman Neural Network. Energy Reports. 2022. vol. 8. pp. 80–88. DOI: 10.1016/j.egyr.2022.08.024.
29. Nguyen V.-Q., Vu M.-H., Pham V.-T., Tran T.-T. A Deep Learning Approach Based on MLP-Mixer Models for Bearing Fault Diagnosis. Proceedings of International Conference on System Science and Engineering (ICSSE'2023). IEEE, 2023. pp. 16–21. DOI: 10.1109/ICSSE58758.2023.10227218.
30. Khoualdia T., Lakehal A., Chelli Z., Khoualdia K., Nessaib K. Optimized Multi Layer Perceptron Artificial Neural Network Based Fault Diagnosis of Induction Motor Using Vibration Signals. Diagnostyka. 2021. vol. 22. no. 1. pp. 65–74. DOI: 10.29354/diag/133091.
31. Naimi A., Deng J., Shimjith S.R., Arul A.J. Fault Detection and Isolation of a Pressurized Water Reactor based on Neural Network and K-Nearest Neighbor. IEEE Access. 2022. vol. 10. pp. 17113–17121. DOI: 10.1109/ACCESS.2022.3149772.
32. Ma Y., Liu H., Zhu Y., Wang F., Luo Z. The NARX Model-Based System Identification on Nonlinear, Rotor-Bearing Systems. Applied Sciences. 2017. vol. 7. no. 9. pp. 911. DOI: 10.3390/app7090911.

33. Choe H.O., Lee M.H. Artificial Intelligence-based Fault Diagnosis and Prediction for Smart Farm Information and Communication Technology Equipment. Agriculture. 2023. vol. 13. no. 11. DOI: 10.3390/agriculture13112124.
34. Taqvi S.A., Tufa L.D., Zabiri H., Maulud A.S., Uddin F. Fault Detection in Distillation Column Using NARX Neural Network. Neural Computing and Applications. 2020. vol. 32. no. 8. pp. 3503–3519. DOI: 10.1007/s00521-018-3658-z.
35. Mustafaraj G., Lowry G., Chen J. Prediction of Room Temperature and Relative Humidity by Autoregressive Linear and Nonlinear Neural Network Models for an Open Office. Energy and Buildings. 2011. vol. 43. no. 6. pp. 1452–1460. DOI: 10.1016/j.enbuild.2011.02.007.
36. Sani M.G., Abdul Wahab N., Sam Y.M., Samsudin S.I., Jamaludin I.W. Comparison of NARX Neural Network and Classical Modelling Approaches. Applied mechanics and Materials. 2014. vol. 554. pp. 360–365. DOI: 10.4028/www.scientific.net/AMM.554.360.
37. Lin Y.P. ARX and NARX Modeling and Control of a Continuous UV/H2O2 Photoreactor for the Aqueous PVA Degradation. Master's Thesis. Toronto Metropolitan University, 2024. 171 p. DOI: 10.32920/25761546.
38. Ling T.G., Rahmat M.F., Husain A.R. A Comparative Study of Linear ARX and Nonlinear ANFIS Modeling of an Electro-hydraulic Actuator System. Jurnal Teknologi (Sciences & Engineering). 2014. vol. 67. no. 5. pp. 1–8. DOI: 10.11113/jt.v67.2833.
39. Renteria-Mena J.B., Plaza D., Giraldo E. Multivariable NARX based Neural Networks Models for Short-term Water Level Forecasting. Engineering Proceedings. 2023. vol. 39. no. 1. DOI: 10.3390/engproc2023039060.
40. Delcroix B., Ny J.L., Bernier M., Azam M., Qu B., Venne J.S. Autoregressive Neural Networks with Exogenous Variables for Indoor Temperature Prediction in Buildings. Building simulation. 2021. vol. 14. no. 1. pp. 165–178. DOI: 10.1007/s12273-019-0597-2.
41. Ramirez C., Acuna G. Forecasting Cash Demand in ATM using Neural Networks and Least Square Support Vector Machine. Proceedings of 16th Iberoamerican Congress on Pattern Recognition (CIARP). Springer Berlin Heidelberg, 2011. pp. 515–522. DOI: 10.1007/978-3-642-25085-9_61.
42. Shao Y., Zhao L. NARX-GA-Elman Method for Mach Number Prediction of Wind Tunnel Flow Field. Instrumentation. 2023. vol. 10. no. 4. DOI: 10.15878/j.cnki.instrumentation.2023.04.004.
43. Nikentari N., Wei H.L. Tide Level Prediction Using NARX-based Recurrent Neural Networks. Proceeding of 27th International Conference on Automation and Computing (ICAC). IEEE, 2022. pp. 1–6. DOI: 10.1109/ICAC55051.2022.9911163.
44. Freitas V.C.G.D., Araujo V.G.D., Crisostomo D.C.D.C., Lima G.F.D., Neto A.D.D., Salazar A.O. Velocity Prediction of a Pipeline Inspection Gauge (PIG) with Machine Learning. Sensors. 2022. vol. 22. no. 23. DOI: 10.3390/s22239162.
45. Ghate V.N., Dudul S.V. Optimal MLP Neural Network Classifier for Fault Detection of Three Phase Induction Motor. Expert Systems with Applications. 2010. vol. 37. no. 4. pp. 3468–3481. DOI: 10.1016/j.eswa.2009.10.041.
46. Teler K., Skowron M., Orłowska-Kowalska T. Implementation of MLP-Based Classifier of Current Sensor Faults in Vector-Controlled Induction Motor Drive. IEEE Transactions on Industrial Informatics. 2024. vol. 20. no. 4. pp. 5702–5713. DOI: 10.1109/TII.2023.3336348.
47. Lin W., Yang C., Lin J., Tsay M.A. Fault Classification Method by RBF Neural Network with OLS Learning Procedure. IEEE Power Engineering Review. 2001. vol. 21. no. 8. DOI: 10.1109/MPER.2001.4311561.
48. Yang P., Wang T., Yang H., Meng C., Zhang H., Cheng L. The Performance of Electronic Current Transformer Fault Diagnosis Model: Using an Improved Whale

Optimization Algorithm and RBF Neural Network. Electronics. 2023. vol. 12. no. 4. DOI: 10.3390/electronics12041066.
49. Jeong K., Choi S.B., Choi H. Sensor Fault Detection and Isolation Using a Support Vector Machine for Vehicle Suspension Systems. IEEE Transactions on Vehicular Technology. 2020. vol. 69. no. 4. pp. 3852–3863. DOI: 10.1109/TVT.2020.2977353.
50. Ramdani O., Beddek K., Haddouche R., Zerrougui M., Chouider N. SVM-Based Approach Fault Detection for PMSG-Wind Energy Conversion System. Journal of Engineering Research. 2025. vol. 13. no. 4. pp. 3378–3393. DOI: 10.1016/j.jer.2025.01.001.
51. Zhou N., Shao Q., Zhou J., Changjie H. Fault Classification of Proton Exchange Membrane Fuel Cells for Vehicles based on XGBoost. Proceedings of 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2021. pp. 1054–1058. DOI: 10.1109/ICBAIE52039.2021.9389943.
52. Hasan S., Toufikuzzaman M. Fault Occurrence Detection and Classification of Fault Type in Electrical Power Transmission Line with Machine Learning Algorithms. International Journal on Electrical Engineering and Informatics. 2022. vol. 14. no. 3. DOI: 10.15676/ijeei.2022.14.3.9.
53. Mehta A., Goyal D., Choudhary A., Pabla B.S., Belghith S. Machine Learning-Based Fault Diagnosis of Self-Aligning Bearings for Rotating Machinery Using Infrared Thermography. Mathematical Problems in Engineering. 2021. vol. 2021. no. 1. DOI: 10.1155/2021/9947300.
54. Shaikh F.A., Kamboh M.Z., Alvi B.A., Khan S., Khan F.M. Condition-Based Health Monitoring of Electrical Machines Using DWT and LDA Classifier. Sir Syed University Research Journal of Engineering and Technology. 2022. vol. 12. no. 2. pp. 95–100. DOI: 10.33317/ssurj.513.
55. Altayef E., Anayi F., Packianather M., Benmahamed Y., Kherif O. Detection and Classification of Lamination Faults in A 15 kVA Three-Phase Transformer Core using SVM, KNN and DT Algorithms. IEEE Access. 2022. vol. 10. pp. 50925–50932. DOI: 10.1109/ACCESS.2022.3174359.
56. Shinde P.V., Desavale R.G., Jadhav P.M., Sawant S.H. A Multi Fault Classification in a Rotor-Bearing System using Machine Learning Approach. Journal of the Brazilian Society of Mechanical Sciences and Engineering. 2023. vol. 45. no. 2. DOI: 10.1007/s40430-023-04015-1.
57. Rakhmawati R., Murdianto F.D., Luthfi A., Rahman A.Y. Thermal Optimization on Incubator using Fuzzy Inference System based IoT. Proceedings of International Conference of Artificial Intelligence and Information Technology (ICAIIT). IEEE, 2019. pp. 464–468. DOI: 10.1109/ICAIIT.2019.8834530.
58. Febriani A., Wahyuni R., Mardeni M., Irawan Y., Melyanti R. Improved Hybrid Machine and Deep Learning Model for Optimization of Smart Egg Incubator. Journal of Applied Data Sciences. 2024. vol. 5. no. 3. pp. 1052–1068. DOI: 10.47738/jads.v5i3.304.
59. Easwaran A., Arvindan P., Dhanyasree E., Surya R., Selvakumar S. Internet of Things Enabled Smart Animal Farm Prototype. Journal of Physics: Conference Series. 2021. vol. 2070. no. 1. DOI: 10.1088/1742-6596/2070/1/012115.
60. Guzman-Zabala J.V., Castro-Martin A.P. Smart and Semi-industrial Egg Incubator with Remote Monitoring Using LoRa Technology. Proceedings of International Conference on Computer Science, Electronics and Industrial Engineering (CSEI). Springer Nature Switzerland, 2024. pp. 522–540. DOI: 10.1007/978-3-031-70981-4_35.
61. Roshanghiyasi H., Ahmad A.H., Hosseinpour S., Jafari A., Mousazadeh H., Asadollahzadeh A. Monitoring and Controlling the Chicken Incubation Process Using

the Internet of Things System. Journal of Agricultural Mechanization. 2022. vol. 7. no. 1. pp. 73–77. DOI: 10.22034/jam.2022.15710.
62. Radhakrishnan K., Jose N., Sanjay S.G., Cherian T., Vishnu K.R. Design and Implementation of a Fully Automated Egg Incubator. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2014. vol. 3. no. 2. pp. 7686–7690.
63. Uzoigwe L.O., Ekezie J.C. Egg Incubator Control System with Short Message Service (sms) Fault Analysis Alert. Journal of Agriculture and Food Sciences. 2013. vol. 11. no. 2. pp. 45–68. DOI: 10.4314/jafs.v11i2.5.
64. Stull R. Practical Meteorology: An Algebra-based Survey of Atmospheric Science. University of British Columbia, 2017. 940 p.
65. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Representations by Back-Propagating Errors. Nature. 1986. vol. 323. no. 6088. pp. 533–536. DOI: 10.1038/323533a0.
66. Elman J.L. Finding Structure in Time. Cognitive science. 1990. vol. 14. no. 2. pp. 179–211. DOI: 10.1207/s15516709cog1402_1.
67. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural computation. 1997. vol. 9. no. 8. pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
68. Cho K., Van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014. pp. 1724–1734. DOI: 10.3115/v1/D14-1179.
69. MacKay D.J.C. Bayesian Interpolation. Neural Computation. 1992. vol. 4. no. 3. pp. 415–447. DOI: 10.1162/neco.1992.4.3.415.
70. Londhe A.S., Ingale A.S., Khadse C.B. Bayesian Regularization Neural Network-based Fault Detection System in HVDC Transmission System. Proceedings of International Conference on Smart Technologies for Energy, Environment, and Sustainable Development (ICSTEESD). Springer Nature, 2022. pp. 601–607. DOI: 10.1007/978-981-16-6875-3_48.
71. Hassan M.S., Kamal K., Ratlamwala T.A.H. Fault Classification of Power Plants using Artificial Neural Network. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. 2022. vol. 44. no. 3. pp. 7665–7680. DOI: 10.1080/15567036.2022.2113936.
72. Liu D.C., Nocedal J. On the Limited Memory BFGS Method for Large Scale Optimization. Mathematical Programming. 1989. vol. 45. no. 1. pp. 503–528. DOI: 10.1007/BF01589116
73. Ljung L. System Identification: Theory for the User. Prentice Hall, 1999. pp. 640.

Атмане Мохамед Амин — аспирант, кафедра автоматизации/факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: обнаружение и изоляция неисправностей, искусственный интеллект, автоматизация. Число научных публикаций — 1. mohamed.atmane@univ-sba.dz; PB 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +(213)4870-0138.

Затла Хишам — Ph.D., старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: автоматическое управление, искусственный интеллект, человеко-машинные системы. Число научных публикаций — 8. hicham.zatla@univ-sba.dz; PB 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +(213)4870-0138.

Толби Билал — старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: искусственный интеллект, гибридные системы, управление системами. Число научных публикаций — 15. bilal.tolbi@univ-sba.dz; РВ 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +213(4870)01-38.

Нуар Суад Фадила — старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: моделирование физических систем, численное моделирование. Число научных публикаций — 2. souadfadila.nouar@univ-sba.dz; РВ 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +213(4870)01-38.

Бухамама Мохамед — старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: электротехника, экологические технологии. Число научных публикаций — 14. mohammed.bouhamama@univ-sba.dz; РВ 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +213(4870)01-38.

Поддержка исследований. Исследование проводится при поддержке Генерального управления по научным исследованиям и технологическому развитию (DGRSDT).

DOI 10.15622/ia.25.2.7

V.-H. Bui, D.-N. Tran, T.-H. Dao, H.Q. Trung, P.V. Kien,
N.V. Thang, D.-T. Tran

IDRRS: IOT INERTIAL DEVICE FOR REAL-TIME ROAD SURFACE CLASSIFICATION AND POSITION ESTIMATION ENHANCEMENT

Viet-Hoan Bui, Duc-Nghia Tran, To-Hieu Dao, Hoang Quang Trung, Pham Vu Kien, Nguyen Van Thang, and Duc-Tan Tran. **IDRRS: IoT Inertial Device for Real-Time Road Surface Classification and Position Estimation Enhancement.**

Abstract. Road surface condition monitoring is essential for enhancing transportation safety and infrastructure maintenance. This study develops an IoT-oriented inertial sensing framework for real-time pavement classification, pothole detection, and enhanced vehicle position estimation. The framework integrates a memory-constrained XGBoost model designed for microcontroller deployment, a velocity-aided GPS interpolation procedure, and an abnormality-index-based pothole detection algorithm. Experimental results on a private dataset and the PVS dataset show classification accuracies of 95.39% and 93.21%, respectively. To examine transferability, the configuration tuned on the private dataset was applied to the PVS dataset without retraining and achieved 92.45% accuracy. Furthermore, the GPS interpolation procedure reduces mean localization errors from 5.571-11.893 m to 1.835–3.563 m across vehicle speeds of 20–50 km/h. An additional contribution of this study is the release of a private dataset capturing vibration signatures from representative road types, supporting further research in road surface classification.

Keywords: Inertial sensor, real-time, microcontroller, machine learning, IoT.

1. Introduction. In recent years, the development of intelligent transportation systems (ITS) [1] and their sub-applications, such as advanced driver assistance systems [2], has significantly increased the demand for diverse and reliable situational data from the traffic environment [3-6]. The core objective of these systems is to enhance road safety, improve traffic management efficiency, optimize the operational performance of vehicles, and ensure passenger comfort [5, 7-9]. To achieve these goals, autonomous and semi-autonomous vehicles require accurate and real-time environmental perception, wherein understanding the characteristics of the road surface is a foundational requirement [10].

Among situational data types, road surface classification (RSC) is considered one of the most critical pieces of information [5, 6]. The ability to accurately distinguish between road surface types – such as asphalt, cobblestone, and dirt – has a direct impact on vehicle dynamics, including grip, braking distance, and vibration patterns [10]. Real-time RSC information is a vital input for adaptive control systems, for example, anti-lock braking systems [11] and traction control systems [12], allowing them to adjust operational parameters to ensure maximum safety [5, 8].

Many methods have been proposed to address the RSC problem [10, 13-15]. Vision-based methods (using cameras), while common, suffer significant

performance degradation due to environmental factors such as changing light conditions, shadows, or adverse weather (rain, snow) [16-19]. 3D scanning methods, such as those using LiDAR, provide high-precision data but require expensive hardware and consume significant computational resources, making them unsuitable for large-scale deployment [5, 7, 8, 20].

Consequently, methods based on inertial sensors (inertial measurement units - IMU), including accelerometers and gyroscopes, have emerged as an effective alternative [5, 13, 20]. These sensors are low-cost, highly durable, and, most importantly, they operate independently of light and weather conditions [5, 8, 13, 14, 20]. Furthermore, they offer a fundamental advantage: instead of passively observing the surface, they directly measure the kinetic response and vibration patterns of the vehicle itself as it interacts with the road [5, 8, 14, 20].

Despite these advantages, achieving robust and scalable RSC from inertial data remains challenging. Vibro-kinetic responses vary significantly across vehicles, suspension systems, driving behaviors, road environments, and sensor mounting positions [4, 7, 13, 14]. These contextual variations introduce strong dependency factors that can limit the generalization capability of machine learning (ML) models. While previous studies have explored various architectures, two primary gaps remain: (i) the lack of evaluation under standardized, context-aware protocol frameworks; (ii) the high computational demand of modern deep learning (DL) models, which limits their deployment on resource-constrained microcontrollers (MCU) [9, 15, 21, 22].

In response to these challenges, this work develops a lightweight and deployable classical machine learning (CML) pipeline. Unlike many previous studies that prioritize accuracy through complex DL models, this study focuses on practical hardware efficiency without compromising generalization. The contributions are fourfold:

– This study introduces the IoT Inertial Device for Real-Time Road Surface Classification and Position Estimation Enhancement (IDRRS). An Extreme Gradient Boosting (XGBoost) classification model is deployed directly onto the device, integrated with a GPS interpolation algorithm. This approach provides road surface condition data and corresponding positioning, achieving classification accuracies of 95.39% on the private dataset and 93.21% on the PVS dataset, respectively.

– Integration of a GPS interpolation algorithm that reduces positioning deviation by 4-5 times compared to conventional GPS.

– A pothole-detection method using an abnormality index computed from triaxial accelerometer signals to localize surface disturbances on reconstructed trajectories.

– A private dataset was constructed based on inertial sensors and positioning data to support feature engineering analysis and the development of predictive models.

2. Related works. The classification of road surface types using inertial sensors and ML is a critical component of ITS, enhancing safety and efficiency. This approach is crucial for enhancing road safety and maintenance by providing real-time data on road conditions [5, 7-10]. Various ML models, including DL and traditional algorithms, have been employed to classify road surfaces based on data from IMU and GPS sensors [10, 13-15]. While DL models offer automated feature discovery, CML methods remain competitive due to their lower computational footprint on embedded systems.

The author [4] established a foundation for modern IMU-based RSC research by introducing the PVS dataset. Comparison between CML and DL methods showed that a convolutional neural network (CNN) architecture achieved 93.17% accuracy, outperforming CML models such as k-nearest neighbors (74.79%). This result was achieved through a 3-experiment split evaluation protocol designed to test model generalization across unseen vehicle, driver, and environmental contexts.

Subsequent research has increasingly focused on improving classification accuracy through more sophisticated DL architectures. A common characteristic of many of these studies is the use of simplified evaluation procedures, such as random 80/20 splits or k-fold cross-validation (CV), rather than the context-aware three-scenario evaluation protocol defined in [4]. These simplified splits allow data from the same vehicle, driver, or scenario to appear in both training and test sets, increasing the risk of context leakage [4] and thus preventing standardized performance comparisons [9].

One early attempt to enhance performance was proposed by the authors in [5], who introduced the SE-ResNet architecture, a residual CNN augmented with Squeeze-and-Excitation (SE) modules. The reported accuracy reached 98.41% using 5-fold CV on the entire dataset. As a result, the comparison with the baseline should be interpreted with caution.

Recurrent neural network (RNN) variants have also been explored to address temporal dependencies in inertial signals. In paper [6] proposed ResBiGRU-SE, a hybrid architecture combining residual connections, bidirectional gated recurrent units (BiGRU), and SE modules. The reported accuracy reached 98.41% through 5-fold CV, making it difficult to assess stability across different vehicles, drivers, and environments.

In paper [8] proposed a hybrid architecture combining CNN and long short-term memory (LSTM) branches to leverage multiple data representations. The model utilizes frequency-domain data for the CNN component and stacked

discrete wavelet transform coefficients for the LSTM branch. On the PVS dataset, this hybrid approach reported an accuracy of 94.78%. However, the evaluation was restricted to a specific 66/34 split (only Experiment 3), providing limited evidence of the model's robustness against independent contextual shifts.

In paper [23] applied Transformer-based architecture to the task, using multi-head attention to model long-range temporal dependencies. The method incorporated a Random Forest (RF)-based feature selection step to reduce input dimensionality. Using an 80/20 split on PVS 1–8, the model achieved a weighted $F1$-score of 97%. When evaluated on an unseen context (PVS 9), performance decreased to 80.41%, illustrating the sensitivity of such models to contextual variation.

In study [24] extended this line of work by proposing a VGG-inspired 1D-CNN architecture achieving 99.3% accuracy with 101.6k parameters. This result was obtained using a simple 80/20 split. Therefore, while the architecture is compact and efficient, its generalization capability under unseen operational conditions remains unverified.

A parallel line of research has focused on improving practicality and computational efficiency. The performance sensitivity observed in several DL models, together with the substantial drop reported by the authors in [23] when evaluated on an unseen context, further underscores the need for approaches that maintain accuracy while operating reliably across diverse conditions and hardware constraints.

Paper [9] are among the few studies that apply the three-scenario evaluation protocol of the author in [4]. Their work introduced two lightweight CNN architectures, SepRNet-1D and SepSERNet-1D, designed with principles inspired by MobileNet V3 through the use of separable convolutions and residual connections. SepRNet-1D achieved an average accuracy of 94.69% with a reported inference time of under 4 ms using a TensorFlow Lite implementation on a CPU-based desktop system (13 GB RAM).

In paper [20] the author revisited tree-based ML models to re-evaluate their potential for RSC. The authors examined RF, Gradient Boosting (GB), and XGBoost, and proposed a Hybrid Filter-Wrapper (HFW) feature selection strategy that combines the efficiency of filter-based ranking with the effectiveness of wrapper-based selection. Their highest reported accuracy was 94.2%, obtained using an XGBoost model with a substantially reduced feature subset, shrinking the original 126 features to 30 through HFW-GB. However, this result was achieved using a 70/30 split, providing limited evidence of the model's robustness against independent contextual shifts.

In summary, while recent advancements in DL and Transformer-based models have increased classification accuracy, these approaches often involve high computational overhead and limited robustness in unseen environments. Furthermore, existing literature frequently overlooks the interplay between classification performance and spatial localization accuracy. The proposed IDRRS device addresses these limitations by integrating a hardware-optimized XGBoost pipeline with a velocity-aided GPS interpolation strategy. This integration achieves competitive classification performance on resource-constrained hardware while improving positioning accuracy in urban settings.

3. Materials and methods.

3.1. Proposed system. The proposed IoT-oriented framework, illustrated in Figure 1, is a compact device designed for real-time RSC and pothole detection on degraded pavements. The hardware architecture, detailed in Table 1, is centered around an ESP32 MCU that serves as the primary processing unit. This MCU executes embedded CML models and anomaly detection algorithms on vibration data acquired from a GY-85 9-axis IMU. Localization is supported by an ATGM336H GPS module, while a SIMCom A7680C modem facilitates data transmission over the LTE network. The sensor node is housed in a 7.3 × 4.5 × 3.4 cm^3 protective casing and is primarily powered by the vehicle's battery system, with an integrated 3.7 V–2000 mAh Li-Po battery serving as a backup to ensure uninterrupted operation during power fluctuations.

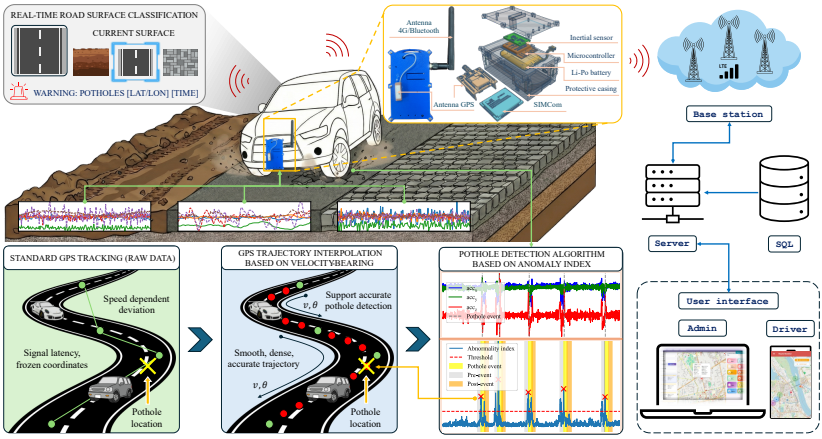


Fig. 1. Architecture of the proposed IoT-based road surface monitoring and alert system

The deployment involves mounting the device on a fleet of vehicles dedicated to road inspection tasks. To capture stable vibration patterns, the sensor node is rigidly mounted at the geometric center of each vehicle, typically near the differential system, with a coordinate convention where the x-axis is oriented leftward, the y-axis forward, and the z-axis downward. To mitigate inertial sensor bias and account for vehicle-specific idle vibrations caused by engine and suspension differences, a calibration procedure is executed whenever the vehicle remains stationary for one second. This process calculates the zero-point offset for the accelerometer and the static bias for the gyroscope. Additionally, the system utilizes the internal thermometer of the ITG3200 for thermal compensation of the inertial readings. The protective casing is also equipped with waterproof gaskets and internal electronic potting to protect the circuitry from humidity and temperature fluctuations.

Table 1. Hardware configuration and experimental platform

Module	**Component**	**Specification**	**Unit**	f_s **(Hz)**
Sensor node	ESP32	520 kB RAM, 448 kB ROM, 4 MB Flash	-	-
	GY-85 9-axis IMU	Accelerometer ADXL345	g (1 g = 9.8 m/s^2)	100
		Gyroscope ITG3200	rad/s	
		Magnetometer HMC5883L	μT	
	GPS ATGM336H	Speed	km/h	1
		Latitude, longitude	Decimal degrees	
	SIMCom A7680C	4G LTE Cat 1	-	-
	Li-Po battery	3.7 V–2000 mAh	-	-
Server HTTP	Compute node	2 × Intel Xeon Gold 6226R (32 cores)	-	-
	Memory	384 GB DDR4 RAM	-	-
	Operating system & Access	AlmaLinux 8, Remote SSH (Visual Studio Code)	-	-

As depicted in the architecture diagram (Figure 1), initial processing occurs locally on the ESP32. Raw sensor signals are segmented and converted into road surface categories while simultaneously being monitored for sudden disturbances using an abnormality-index-based algorithm. Because standard GPS modules often suffer from update delays or signal loss in challenging environments, a position interpolation procedure is integrated to provide continuous and enhanced location estimation for detected anomalies. The processed results are then transmitted via the cellular network to an SQL database on a central server. This centralized repository supports a two-part user interface: an administrator dashboard for traffic authorities to visualize pavement quality and prioritize repairs, and a driver interface that delivers real-time hazard warnings. This integrated flow facilitates a proactive approach to road maintenance and enhances operational safety through timely anomaly reporting.

3.2. Dataset.

3.2.1. Private dataset. The private dataset[1] was collected using a fleet of vehicles dedicated to regional road inspection tasks. Asphalt, cobblestone, and dirt were selected as the target surface categories because they constitute the predominant pavement types in regional transportation networks and represent the primary focus of contemporary road condition monitoring studies [4,20,25].

Data collection followed a controlled experimental protocol where each session was dedicated to a single road surface type under the direct supervision of two researchers. These supervisors observed the conditions and ensured the recorded data corresponded precisely to the targeted pavement category, facilitating ground-truth annotation at the source without the requirement for post-processing software tools.

The acquired signals were stored on a card in plain-text format, with a sampling frequency of 100 Hz for the inertial and magnetometer sensors and 1 Hz for the GPS module. Each data file contained triaxial accelerometer, gyroscope, and magnetometer measurements, alongside GPS latitude, longitude, timestamps, and speed. Figure 2 provides the GPS trajectories and representative photographs of the sampled road surfaces to offer visual context for the experimental environments. In total, the recordings amounted to 903.90 minutes (15.06 hours), corresponding to approximately 196 MB of raw measurements.

Fig. 2. GPS trajectories for dirt (red), cobblestone (blue), and asphalt (green) segments (top) and representative images of the three surface types (bottom)

3.2.2. Public dataset. The public dataset used in this study is the PVS[2] benchmark [4], which complements the private dataset by providing a

[1] https://dx.doi.org/10.21227/x6av-va13

[2] https://github.com/jefmenegazzo/Intelligent-Vehicle-Perception-Based-on-Inertial-Sensing-and-Artificial-Intelligence

structured basis for evaluating reproducibility and generalization under diverse contextual factors. The dataset comprises nine sub-datasets (PVS 1–PVS 9) collected from three vehicles driven by three independent drivers across three road scenarios, with all samples labeled into dirt, cobblestone, and asphalt classes. Each vehicle carries MPU-9250 modules installed at three locations – dashboard (DB), above suspension (AS), and below suspension (BS) – on both left and right sides of the front axle. To ensure valid generalization, this study adheres to the 3-experiment split evaluation protocol proposed by the author in [4], a method designed to test robustness against changes in vehicle type, driver behavior, and environmental context [9].

3.3. Windowing. Continuous sensor signals are segmented into fixed-length windows using a sliding window of 600 samples (6 seconds at 100 Hz) with 50% overlap. This configuration is selected to balance classification performance with real-time operational constraints. Regarding window duration, research on inertial sensor data indicates that recognition accuracy tends to saturate at sizes around 6 seconds, as this interval provides sufficient temporal coverage to capture essential cyclic characteristics of the movement [26]. Specifically, in the context of RSC for MCU deployment, a 6-second window has been shown to effectively balance high accuracy with the limited processing capabilities of low-end hardware [25]. Furthermore, the 50% overlap results in a 300-sample stride, ensuring that new predictions are generated every 3 seconds. This setup complies with established real-time latency requirements, where a prediction delay of no more than 3 seconds is considered acceptable for timely monitoring [4, 25]. The general characteristics of the datasets and the distribution of the classification instances are summarized in Table 2.

Table 2. Summary of dataset characteristics and instances distribution

Dataset	Class	Instances (number of window segments)	Total instances	Split protocol
Private	Dirt road Cobblestone road Asphalt road	3258 2157 3624	9039	10×10-fold NCV
Public PVS	Dirt road Cobblestone road Asphalt road	1996 2096 3088	7180	3-scenario split [4, 9]

For the private dataset, the segmentation process yielded 9039 windows, which are utilized in a 10×10-fold nested cross-validation (NCV) procedure. Regarding the PVS dataset, a total of 7180 unique windows are generated (Table 3). Performance evaluation on the PVS dataset follows a protocol involving three independent experiments (Exp. 1-3) to ensure generalization across different vehicles, drivers, and scenarios [4,9]. Within this three-scenario

framework, the aggregate evaluation comprises 21540 processed instances, consisting of 14146 training windows and 7394 testing windows (Table 4).

Table 3. Distribution of segmented windows per PVS dataset

Dataset	Vehicle	Driver	Scenario	Number of windows		
				Dirt	Cobblestone	Asphalt
PVS 1	Volkswagen Saveiro	1	1	172	412	374
PVS 2	Volkswagen Saveiro	1	2	298	138	392
PVS 3	Volkswagen Saveiro	1	3	190	176	336
PVS 4	Fiat Bravo	2	1	160	384	336
PVS 5	Fiat Bravo	2	2	404	120	366
PVS 6	Fiat Bravo	2	3	160	210	268
PVS 7	Fiat Palio	3	1	158	362	334
PVS 8	Fiat Palio	3	2	300	126	396
PVS 9	Fiat Palio	3	3	154	168	286
Total				1996	2096	3088

Table 4. Train/test split distribution of segmented windows for the three evaluation experiments on the PVS dataset

	Dirt road		Cobblestone road		Asphalt road	
	Train	Test	Train	Test	Train	Test
Exp. 1	994	1002	1712	384	1934	1154
Exp. 2	1272	724	1382	714	2118	970
Exp. 3	1244	752	1438	658	2052	1036

3.4. Feature engineering. A feature engineering stage was applied to transform each segmented window into a compact representation for ML input. As the design targets real-time operation on resource-constrained hardware, only low-complexity time-domain features were considered, while frequency-domain descriptors were excluded due to their high computational complexity [21]. For each inertial channel, a comprehensive set of 13 statistical descriptors was selected to characterize signals across four dimensions: central tendency (mean, median), dispersion (standard deviation–SD, root mean square–RMS, interquartile range–IQR, maximum, minimum, and range), distribution shape (skewness, kurtosis), and temporal dynamics (Hjorth mobility, Hjorth complexity, and autocorrelation). Additionally, the mean and SD of the speed signal were included, as vibration intensity correlates with vehicle velocity [4]. This specific ensemble was designed to provide a multidimensional characterization of vibration patterns while maintaining compatibility with the computational constraints of embedded inference [20, 21].

3.5. Constrained hyperparameter optimization. To optimize the models for deployment on memory-limited MCUs, two primary hyperparameters – `n_estimators` and `max_depth` – were systematically

tuned within a search space of 1–50 and 1–20, respectively. These specific parameters were selected because they exert a dominant influence on both the classification performance and the final memory footprint of tree-based ensembles [21, 27, 28]. Other hyperparameters were maintained at their default settings as defined in the ML libraries, with the `random_state` fixed at 42 to ensure experimental reproducibility. All four tree-based classifier families (XGBoost, RF, Light Gradient Boosting Machine–LGBM, and Extra Trees–ET) were evaluated under this same constrained search space. For each candidate configuration, the model was fitted on the entire dataset and exported to a C/C++ header file using the `micromlgen` or `m2cgen` library [22].

The private dataset was collected using a dedicated inspection fleet operating under homogeneous mechanical conditions and consistent driving routines. Because the data do not exhibit strong driver–vehicle–scenario dependency factors, group-aware or experiment-aware resampling is unnecessary. Hyperparameter optimization (HPO) on this dataset therefore proceeds in two stages. First, a 10-fold flat cross-validation (FCV) is applied to enumerate the constrained search space, evaluate the relative ranking of the candidate configurations, and filter out models whose exported headers exceed the memory constraint. Second, only the deployable configurations identified during FCV are assessed through a full 10×10-fold NCV procedure. FCV determines the final deployable hyperparameter configuration for each classifier family, whereas NCV yields an unbiased estimate of the generalization performance of the model trained with that configuration.

Although formal HPO is conducted only on the private dataset, the PVS dataset features a structurally different evaluation protocol consisting of three experiment-wise splits. To ensure that the proposed pipeline remains reliable under this context-aware partitioning, the same constrained search space is applied within each PVS training split solely for the purpose of examining the stability of the overall pipeline under heterogeneous operational conditions. No information from the PVS test sets is accessed during this process, and the application of the search space on PVS functions as a robustness check to verify that the proposed methodology maintains consistent behavior across distinct data-partitioning structures.

3.6. Velocity-aided GPS interpolation. During data acquisition, GPS receivers often operate at 1 Hz or occasionally fail to provide updates due to satellite occlusion and multipath effects. This data sparsity creates a resolution gap in the recorded trajectory, degrading the temporal alignment with the 100 Hz IMU stream. Consequently, these artifacts reduce the reliability of the positional data associated with road surface types and localized surface anomalies.

To mitigate these issues, a velocity-aided interpolation procedure is applied to reconstruct geographically consistent intermediate positions. Unlike generic polynomial interpolation, this method utilizes the physical motion parameters of the vehicle. The procedure assumes short-term continuous motion and estimates the travelled distance S from the instantaneous speed v and elapsed time t between two valid GPS fixes:

$$S = v \times t. \tag{1}$$

A great-circle navigation model is employed with an Earth radius $R = 6371000$ m. Given two valid observations (ϕ_1, λ_1) and (ϕ_2, λ_2), the bearing angle θ is computed as:

$$\theta = \operatorname{atan2}\Big(\sin(\Delta\lambda)\cos(\phi_2),\ \cos(\phi_1)\sin(\phi_2) - \sin(\phi_1)\cos(\phi_2)\cos(\Delta\lambda)\Big), \tag{2}$$

with $\Delta\lambda = \lambda_2 - \lambda_1$. The interpolated geographic position $(\mathrm{lat}_2, \mathrm{lon}_2)$ at distance S from $(\mathrm{lat}_1, \mathrm{lon}_1)$ along direction θ is determined as:

$$\begin{cases} \mathrm{lat}_2 & = \arcsin\big(\sin(\mathrm{lat}_1)\cos\ \frac{S}{R} + \cos(\mathrm{lat}_1)\sin\ \frac{S}{R}\cos\theta\big) \\ \mathrm{lon}_2 & = \mathrm{lon}_1 + \arctan 2\big(\sin\theta\ \sin\ \frac{S}{R}\cos(\mathrm{lat}_1),\ \cos\ \frac{S}{R} - \sin(\mathrm{lat}_1)\sin(\mathrm{lat}_2)\big) \end{cases}. \tag{3}$$

The interpolation algorithm scans GPS samples and detects intervals where coordinates remain unchanged within an angular tolerance $\varepsilon = 4.5 \times 10^{-6}$ degrees (approximately 0.5 m). Intermediate points are inserted whenever the instantaneous speed is positive. To ensure synchronization with the inertial sensor data, interpolated positions are generated at $F_s = 100$ Hz. The bearing angle is estimated from the nearest available GPS observations, and since θ is recalculated whenever new observations become available, the reconstructed trajectory follows curved paths rather than imposing piecewise-linear interpolation. Figure 3 illustrates the reconstruction principle.

Algorithm 1 summarizes the complete procedure. This method produces a temporally dense trajectory that reflects t he v ehicle motion more closely than raw GPS logs. By bridging the resolution gap between low-frequency GPS updates and high-frequency IMU sampling, this method establishes the spatial reference required to map road surface conditions.

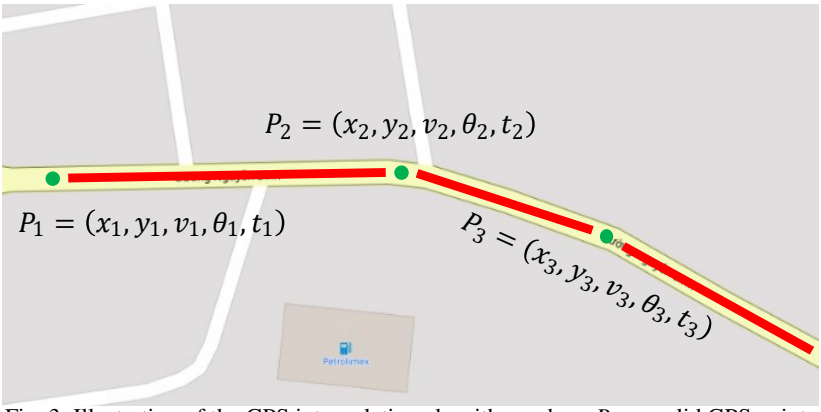


Fig. 3. Illustration of the GPS interpolation algorithm, where P_i are valid GPS points and intermediate points are reconstructed using velocity v and bearing angle θ

Algorithm 1: Velocity-aided GPS interpolation

Input: GPS samples (lat, lon, v, t); interpolation frequency F_s.
Output: Interpolated sequence $\mathscr{L}$.

$step \leftarrow 1/F_s$;
$\mathscr{L} \leftarrow \varnothing$;
$i \leftarrow 1$;
while $i \leqslant N$ **do**
  Find $j > i$ such that GPS$(j) \neq$GPS(i) within tolerance
  Append sample i to $\mathscr{L}$
  if $j > N$ *or* $v_i \leqslant 0$ *or* $j = i+1$ **then**
    $i \leftarrow i+1$; continue
  end
  Choose θ from nearest valid GPS points
  $\Delta T \leftarrow t_j - t_i$
  for *each* Δt *in* $\{step, 2\,step, \ldots, < \Delta T\}$ **do**
    $S \leftarrow v_i \Delta t$;
    $(\hat{lat}, \hat{lon}) \leftarrow$ dest_point$(lat_i, lon_i, \theta, S)$
    Append $(\hat{lat}, \hat{lon}, t_i + \Delta t)$ to $\mathscr{L}$
  end
  Append sample j;
  $i \leftarrow j$;
end
return $\mathscr{L}$

3.7. Pothole detection using abnormality index. Restoring spatial continuity is essential for locating localized road surface anomalies. At an urban vehicle speed of 40 km/h, a standard 1 Hz GPS receiver provides a position update only every 11 m, creating a resolution gap that complicates the localization of potholes. By incorporating the 100 Hz reconstructed path, the system achieves the spatial granularity necessary to map detected potholes onto the estimated vehicle trajectory.

A lightweight abnormality-index method is applied to identify these events from triaxial accelerometer measurements [21]. For each axis, a rolling mean window of size $W = 100$ samples (corresponding to 1 second at $f_s = 100$ Hz) is utilized to filter out low-frequency trends. Let $a_x(n)$, $a_y(n)$, $a_z(n)$ denote the raw accelerations at sample n, and $\bar{a}_x(n)$, $\bar{a}_y(n)$, $\bar{a}_z(n)$ be their corresponding rolling means. The abnormality index (AbI) is defined as:

$$AbI(n) = \sqrt{\left[a_x(n) - \bar{a}_x(n)\right]^2 + \left[a_y(n) - \bar{a}_y(n)\right]^2 + \left[a_z(n) - \bar{a}_z(n)\right]^2}. \quad (4)$$

A detection threshold (Th) is established based on the statistical distribution of the index:

$$Th = \mu_{\text{AbI}} + k \cdot \sigma_{\text{AbI}}, \quad (5)$$

where k is a scalar that controls the balance between sensitivity and robustness. Through experimental analysis, k is set to 2. To prevent redundant detections, samples satisfying $AbI(n) > Th$ are grouped into a single pothole event if they occur within a 1.5-second interval. This interval accounts for the sequential impacts of the front and rear axles on the same surface anomaly, which manifest as double peaks.

Figure 4 illustrates the pothole detection process, including raw accelerometer signals, the abnormality index, and the localization of events along the interpolated trajectory. By leveraging the high-resolution spatial reference from the interpolation algorithm, the system ensures that detected potholes are assigned coordinates consistent with the physical impacts recorded by the IMU. This integration provides the detailed geographic information required for road maintenance and hazard warning.

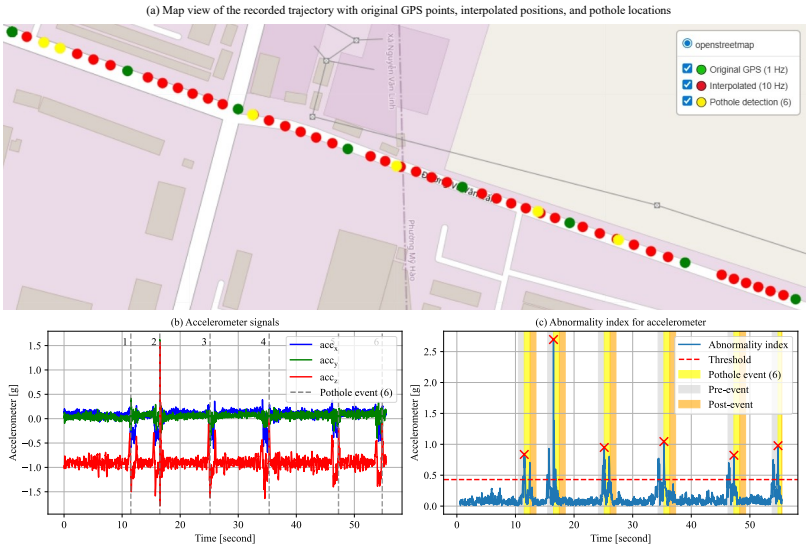


Fig. 4. Detection of pothole events using the abnormality index aligned with interpolated GPS positions

4. Result and discussion.

4.1. Evaluation on the private dataset. Table 5 summarizes the performance and Flash footprint of the four CML model families under the constrained HPO search space on the private dataset. The performance is reported as the mean accuracy $\pm$ SD across the 10-fold NCV, accompanied by the 95% confidence interval (CI). Among the evaluated methods, XGBoost attained the highest accuracy at 95.39 $\pm$ 0.82%, with a 95% CI of [94.80% – 95.97%], demonstrating strong stability across different data splits. LGBM achieved a comparable accuracy of 95.08 $\pm$ 0.56%, while RF and ET yielded lower performance at 92.45% and 91.72%, respectively.

Table 5. Performance and size of optimized CML models on the private dataset

Model	Sensor	Configuration	HPO	Accuracy	95% CI	Flash (kB)
XGBoost	Center	$n_{\text{est}} \in [1, 50]$ $m_{\text{depth}} \in [1, 20]$	{42, 6}	95.39 $\pm$ 0.82	[94.80 - 95.97]	975.79
RF			{13, 12}	92.45 $\pm$ 0.67	[91.97 - 92.94]	1018.04
LGBM			{49, 13}	95.08 $\pm$ 0.56	[94.67 - 95.48]	856.88
ET			{13, 12}	91.72 $\pm$ 0.64	[91.26 - 92.19]	1015.13

To determine whether the performance difference between the two leading models, XGBoost and LGBM, is statistically significant, McNemar's test is conducted (Table 6(a)). The contingency table for McNemar's test

accounts for the counts of correct and incorrect predictions between the two classifiers. For the private dataset, the test yields a χ^2 statistic of 4.1897 with a p-value of 0.0407. Since $p < 0.05$, the performance difference between XGBoost and LGBM is considered statistically significant.

Table 6. Statistical significance comparison between models using McNemar's test

Dataset	Model	LGBM true	LGBM false	χ^2 statistic	p-value
(a) Private (sample size $N = 9039$ windows)	XGB true	8521	101	4.1897	0.0407*
	XGB false	73	344		
(b) Public PVS (sample size $N = 7394$ windows)	XGB true	6744	148	9.3279	0.0023**
	XGB false	99	403		

*$p < 0.05$, **$p < 0.01$ indicates statistical significance levels.

The constraints imposed on the HPO search space are strictly governed by the physical memory limitations of the target ESP32 MCU. According to official specifications from Espressif, the ESP32 features 520 kB RAM, 448 kB ROM, and 4 MB Flash memory (Table 1). In practical implementation, a significant portion of the Flash memory is reserved for system partitions, including the bootloader, Wi-Fi/Bluetooth stacks, and Over-the-Air update buffers. Consequently, the application partition is typically limited to approximately 1.2 MB in standard partition tables. Analysis of the HPO results shows that model configurations exceeding the predefined limits result in header files larger than 1 MB (Figure 5). Such footprints are unsuitable for deployment as they risk exceeding the available application storage. By limiting these parameters, the optimized XGBoost model occupies 975.79 kB, effectively utilizing the Flash budget without compromising system operational integrity.

A detailed characterization of the classification behavior is shown in the per-class report in Table 7 and the confusion matrix in Figure 6(a). For dirt-road segments, XGBoost correctly identifies 3072 out of 3258 windows, corresponding to a recall of 94.29%. Most misclassified cases are assigned to the cobblestone class (184 windows), with very limited confusion involving asphalt (2 windows). Cobblestone represents the most challenging class; its vibration patterns exhibit irregular structures with partial similarity to those of dirt roads. As a result, XGBoost correctly recognizes 1963 out of 2157 cobblestone windows (recall 91.01%), with misclassifications predominantly shifting toward dirt. Asphalt displays the most distinct signature among the three classes and is classified with high reliability: 3587 out of 3624 asphalt windows are correctly predicted, yielding a recall of 98.98% and an $F1$-score of 98.60%.

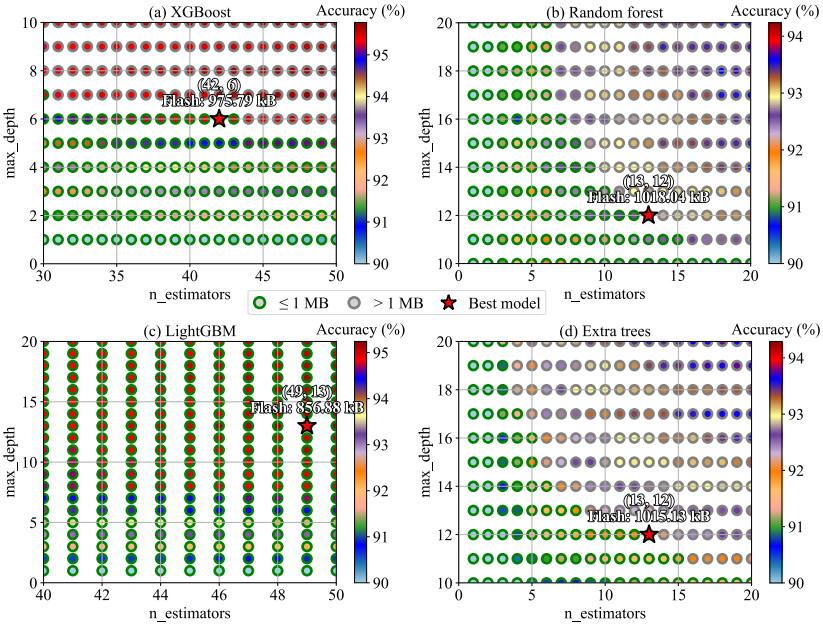


Fig. 5. The correlation between hyperparameter configuration and Flash memory footprint during parameter tuning

Table 7. Classification report of CML models on the private dataset

Model	Class	Precision	Recall	$F1$-score
XGBoost	Dirt road	95.55	94.29	94.92
	Cobblestone road	90.38	91.01	90.69
	Asphalt road	98.22	98.98	98.60
RF	Dirt road	91.75	90.85	91.30
	Cobblestone road	84.32	86.28	85.29
	Asphalt road	98.06	97.57	97.81
LGBM	Dirt road	94.88	94.38	94.63
	Cobblestone road	90.22	89.85	90.03
	Asphalt road	98.11	98.81	98.46
ET	Dirt road	90.31	90.36	90.33
	Cobblestone road	84.00	83.73	83.86
	Asphalt road	97.58	97.71	97.64

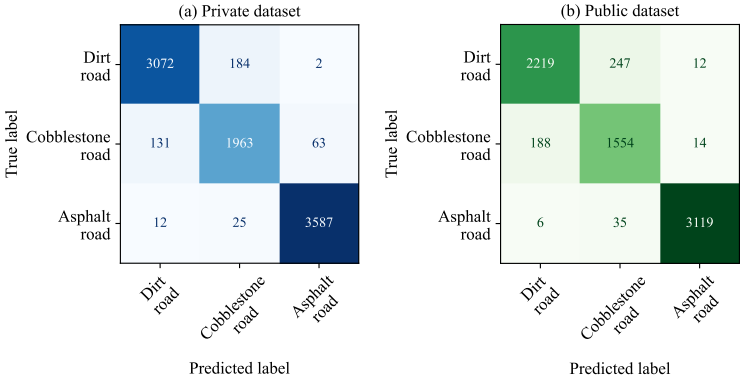


Fig. 6. Confusion matrices of the XGBoost models on: a) the private dataset (pooled 10×10-fold NCV predictions); b) the PVS dataset (combined test sets Exp. 1–3)

4.2. Evaluation on the public dataset.

4.2.1. Impact of sensor placement. A central objective in evaluating the PVS dataset is to examine how different IMU mounting positions influence classification performance. Because the dataset includes three placements – DB, AS, and BS – it enables a systematic assessment of how vibrational richness varies across the vehicle structure. Table 8 reports the average accuracy of the four CML model families across seven sensor configurations.

Table 8. Average accuracy of CML models under different sensor placement configurations on the PVS dataset

Sensor location	Accuracy (%)			
	XGBoost	**RF**	**LGBM**	**ET**
	Single placement			
DB	89.06	82.68	88.78	82.38
AS	90.29	87.62	89.86	84.11
BS	90.90	87.29	91.27	83.97
	Dual placement			
DB, AS	91.21	87.83	89.88	83.51
DB, BS	92.69	87.29	91.72	84.49
AS, BS	**93.21**	**90.73**	**92.56**	**86.50**
	All placements			
DB, AS, BS	93.19	89.68	91.28	85.11

In the single-placement setting, sensors closer to the tire–road interface consistently provide higher accuracy than the cabin-mounted dashboard sensor. Both AS and BS reach accuracy levels around 90%, whereas DB attains 89.06%

with XGBoost. This behavior aligns with mechanical intuition: the suspension system attenuates road-induced vibrations before they propagate to the cabin, reducing the discriminative content available at the dashboard location [4].

For dual-placement configurations, combining AS and BS yields the highest overall performance for all model families, reaching 93.21% with XGBoost. This configuration benefits from capturing two complementary vibration pathways – pre-suspension and post-suspension – thereby preserving a richer representation of road-surface irregularities. Configurations involving the dashboard sensor (DB, AS) or (DB, BS) result in slightly lower accuracy, indicating that the DB channel contributes limited additional information in this context.

The addition of the dashboard sensor to form the three-sensor configuration (DB, AS, BS) does not yield further improvement relative to the dual (AS, BS) setup; XGBoost accuracy decreases marginally from 93.21% to 93.19%. This outcome suggests that the dashboard channel introduces low-informative redundancy that can slightly interfere with decision boundaries established from the more discriminative AS and BS features.

4.2.2. Performance of the models. Having established the optimal sensor configuration, the next step involved evaluating the end-to-end performance of the embedded-oriented pipeline under the three PVS experiments. All evaluations in this subsection relied exclusively on the (AS, BS) dual-sensor configuration.

Table 9 summarizes the performance and Flash footprint of the four CML model families under the constrained HPO search space on the PVS dataset.

Table 9. Performance and size of optimized CML models on the PVS dataset

Model	Sensor	Configuration	HPO	Accuracy (%)	95% CI	Flash (kB)
XGBoost	AS, BS	$n_{\text{est}} \in [1, 50]$ $m_{\text{depth}} \in [1, 20]$	{50, 3}	93.21 ± 0.16	[92.80 - 93.61]	212.88
RF			{20, 9}	90.73 ± 0.52	[89.45 - 92.01]	912.08
LGBM			{45, 5}	92.56 ± 0.49	[91.34 - 93.77]	441.42
ET			{9, 12}	86.50 ± 2.21	[81.00 - 92.00]	915.39

The accuracy is reported as the mean $\pm$ SD across the three PVS experiments, accompanied by the 95% CI. Among the evaluated methods, XGBoost attained the highest overall accuracy, reaching 93.21 ± 0.16% with a 95% CI of [92.80% – 93.61%]. This narrow interval and low SD demonstrated the high stability and generalization capability of the model across different vehicles, drivers, and scenarios. Notably, this performance level aligned with the 93.17% accuracy of the CNN baseline originally reported by the author in [4] while offering a significantly more efficient Flash footprint of 212.88 kB. The remaining models exhibited varying degrees of effectiveness. LGBM

achieved a comparable accuracy of $92.56 \pm 0.49\%$, while RF and ET yielded lower accuracies of 90.73% and 86.50%, respectively.

The statistical advantage of XGBoost is also observed on the PVS dataset (Table 6(b)). The McNemar's test results in a χ^2 statistic of 9.3279 and a p-value of 0.0023. This result confirms the statistical significance of the performance difference ($p < 0.01$), suggesting that the model remains robust despite the higher heterogeneity of the PVS dataset.

To provide a finer-grained view of generalization under the PVS evaluation protocol, the per-class behavior of the four CML models was examined using the aggregated test sets from Exp. 1–3. Table 10 presents precision, recall, and $F1$-score for each road category, and Figure 6(b) visualizes the confusion matrix of the best-performing model (XGBoost), constructed from all 7394 test windows.

Table 10. Classification report of CML models on the PVS dataset

Model	**Class**	**Precision**	**Recall**	$F1$**-score**
XGBoost	Dirt road	91.96	89.55	90.74
	Cobblestone road	84.64	88.50	86.53
	Asphalt road	99.17	98.70	98.94
RF	Dirt road	89.56	84.50	86.96
	Cobblestone road	78.87	85.65	82.12
	Asphalt road	98.76	98.42	98.59
LGBM	Dirt road	91.22	88.10	89.63
	Cobblestone road	82.80	87.70	85.18
	Asphalt road	99.33	98.73	99.03
ET	Dirt road	84.43	76.84	80.46
	Cobblestone road	70.02	79.67	74.53
	Asphalt road	98.57	97.97	98.27

Among all evaluated models, asphalt consistently emerges as the easiest surface type to classify. For XGBoost, 3119 of the 3160 asphalt windows are correctly identified, corresponding to a recall of 98.70% and an $F1$-score of 98.94%. Dirt-road segments display moderately lower, though still robust, generalization. XGBoost attains a recall of 89.55%, correctly classifying 2219 out of 2478 dirt windows. Cobblestone represents the most challenging class for all models. XGBoost reaches an $F1$-score of 86.53% and a recall of 88.50%, correctly identifying 1554 out of 1756 cobblestone windows. The confusion matrix shows that most errors involve confusion with dirt, a phenomenon also noted in prior studies [4, 9].

4.2.3. Robustness across vehicles and scenarios. Table 11 presents the performance metrics under Leave-One-Vehicle-Out (LOVO) and

Leave-One-Scenario-Out (LOSO) CV protocols. These evaluations examine the model sensitivity to variations in vehicles and environmental conditions. Under the LOVO protocol, the model achieves an average accuracy of 89.63% and a macro $F1$-score of 87.18%. The accuracy drop observed for the Fiat Palio (82.49%) indicates that differences in vehicle mechanical responses can influence the recorded vibration data. For the LOSO protocol, the average accuracy and macro $F1$-score are 89.26% and 86.19%, respectively. Scenario 3 represents the most challenging subset with an accuracy of 86.24%, reflecting the influence of varying road profiles or driving environments. The fluctuations in performance across different scenarios and vehicles suggest that domain shifts affect the stability of the classification.

Table 11. Model robustness under LOVO and LOSO CV on the PVS dataset

Training	**Testing**	**Accuracy** (%)	**Macro $F1$-score** (%)
	Leave-One-Vehicle-Out CV		
Volkswagen Saveiro, Fiat Bravo	Fiat Palio	82.49	77.06
Volkswagen Saveiro, Fiat Palio	Fiat Bravo	93.06	92.35
Fiat Bravo, Fiat Palio	Volkswagen Saveiro	93.33	92.12
Average		89.63	87.18
	Leave-One-Scenario-Out CV		
Scenario 1, 2	Scenario 3	86.24	83.23
Scenario 1, 3	Scenario 2	93.39	90.76
Scenario 2, 3	Scenario 1	88.15	84.58
Average		89.26	86.19

In the PVS dataset, Leave-One-Vehicle-Out CV is equivalent to Leave-One-Driver-Out CV.

4.3. Model interpretability and ablation study.

4.3.1. Feature importance and speed sensitivity. The interpretability of the optimized XGBoost model (Fig. 7) is evaluated using SHAP (SHapley Additive exPlanations) values to quantify the contribution of each feature to the classification outcomes. For the asphalt class, low values of `IQR_Acc_Y` and higher `Mean_Speed` are the most influential indicators. For cobblestone roads, high values of `SD_Acc_Y` and lower `Mean_Speed` are significant predictors. In contrast, the classification of dirt roads relies heavily on `IQR_Acc_Y` and low values of `HM_Acc_Y`. These results show that the feature-engineering pipeline effectively captures the physical differences in road-vehicle interactions.

The impact of vehicle speed on model predictions is illustrated in the SHAP dependence plots (Fig. 8). For dirt roads, the model maintains higher confidence over a broader range, up to 40 km/h. Regarding cobblestone roads, the positive contribution is highest at very low speeds (below 25 km/h) and drops significantly as velocity increases. Beyond this threshold, the fluctuating SHAP values suggest that vibration signatures of cobblestone roads can overlap with those of dirt surfaces. Conversely, the asphalt class shows a clear positive correlation with speed. Beyond 50 km/h, the average speed becomes a major

factor for identification, as the surface smoothness allows for higher travel velocities. Therefore, the inclusion of the speed feature provides useful information for surface classification under different driving conditions.

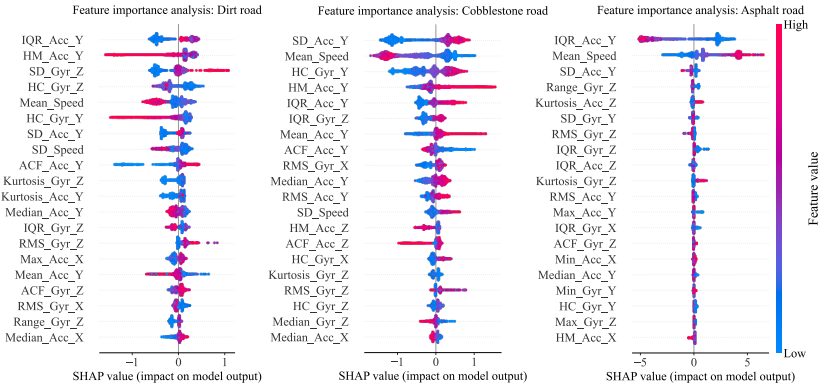


Fig. 7. Feature importance analysis of the XGBoost model using SHAP values

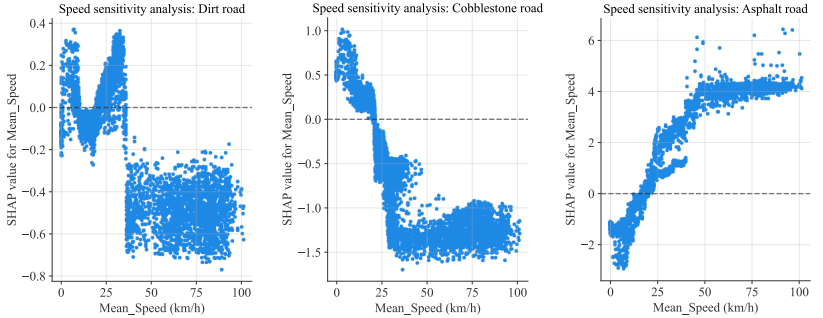


Fig. 8. SHAP dependence plot illustrating the sensitivity of model predictions to vehicle speed variations

4.3.2. Analysis of sensor fusion, feature selection, and window size. The contribution of different system components is evaluated through an ablation study covering sensor configurations, feature sets, and window sizes (Table 12). Dual-sensor fusion (AS, BS) provides the highest accuracy, as single-placement or single-sensor configurations lead to performance drops between 1.83% and 4.24%. Regarding the feature set, removing specific metrics such as range or ACF results in a marginal accuracy increase but leads to wider 95% CIs, suggesting that the proposed 13-feature combination

offers better stability. This feature set also significantly outperforms the combinations proposed in prior studies [4, 20, 25], which show accuracy reductions ranging from 2.54% to 3.32%. Furthermore, excluding the speed feature decreases accuracy by 1.14%. Analysis of window sizes shows that classification performance improves as the duration increases from 1 s to 6 s. Smaller windows fail to capture sufficient vibrational context, while larger windows introduce redundant information and increase identification latency. Therefore, the proposed configuration is effective for real-time RSC, providing a suitable balance between accuracy and operational efficiency.

Table 12. Ablation study investigating the impact of sensor fusion, feature selection, and window sizes on classification accuracy

Ablation category	Configuration variant	Accuracy (%)	Δ (%)	95% CI
Proposed	AS, BS (13 features, 6-second window, 50% overlap)	93.21	-	[92.80 - 93.61]
1. Sensor fusion	Accelerometer only	91.38	-1.83	[90.71 - 92.05]
	Gyroscope only	88.97	-4.24	[86.65 - 91.29]
	Single placement (AS only)	90.29	-2.92	[89.53 - 91.04]
	Single placement (BS only)	90.90	-2.31	[85.91 - 95.89]
2. Feature set	Remove mean	92.43	-0.78	[89.71 - 95.16]
	Remove SD	93.17	-0.04	[92.38 - 93.96]
	Remove RMS	92.86	-0.35	[92.05 - 93.68]
	Remove max	92.82	-0.39	[92.40 - 93.24]
	Remove min	92.94	-0.27	[92.44 - 93.45]
	Remove range	93.25	+0.04	[92.18 - 94.31]
	Remove median	92.99	-0.22	[91.41 - 94.57]
	Remove IQR	92.62	-0.59	[92.06 - 93.18]
	Remove skewness	92.95	-0.26	[91.89 - 94.02]
	Remove kurtosis	92.72	-0.49	[90.37 - 95.07]
	Remove HM	93.17	-0.04	[91.45 - 94.89]
	Remove HC	92.59	-0.62	[91.32 - 93.86]
	Remove ACF	93.23	+0.02	[92.08 - 94.37]
	Without speed feature	92.07	-1.14	[90.83 - 93.30]
	In paper [4] proposed mean, SD, variance	89.89	-3.32	[87.72 - 92.06]
	[20] proposed mean, SD, max, min, range, variance, median	90.67	-2.54	[88.27 - 93.06]
	In paper [25] proposed mean, RMS, SD, variance, median, range	90.24	-2.97	[87.52 - 92.97]
3. Window size	1-second (0% overlap)	87.47	-5.74	[80.91 - 94.03]
	2-second (0% overlap)	89.96	-3.25	[86.42 - 93.49]
	3-second (0% overlap)	90.17	-3.04	[85.25 - 95.09]
	4-second (50% overlap)	91.51	-1.70	[88.25 - 94.78]
	5-second (50% overlap)	92.17	-1.04	[89.65 - 94.69]

Δ *represents the change in accuracy when the model varies parameters.*

4.3.3. Cross-dataset transferability. To examine transferability, the model optimized on the private dataset was applied directly to the PVS dataset without any re-training, hyperparameter adjustment, or experiment-specific modifications. The transferred model attains 92.45% $\pm$ 0.44%, representing a reduction of only 0.76% relative to the PVS-tuned configuration. This limited

degradation, despite the substantial domain shift between datasets, indicates the stability of the proposed 6-second windowing, 13 handcrafted features, and constrained hyperparameter search.

Table 13 compares the performance of the proposed pipeline with existing studies using the PVS dataset. While architectures such as 1D-CNN [24] and Transformers [23] report accuracies exceeding 97%, these results are obtained through 80/20 or k-fold splits, which often do not isolate contextual dependencies [4]. Moreover, the heavy memory footprint and floating-point operations of these models often exceed the hardware limits of low-power MCUs [21]. In contrast, under the 3-scenario protocol, the memory-constrained XGBoost model achieves 93.21%, which is competitive with the 93.17% CNN baseline [4]. The performance is also comparable to modern lightweight architectures such as SepRNet-1D (94.69%) [9], which was evaluated under the same protocol but requires significantly higher computational resources.

Table 13. Comparison of RSC methods on the PVS dataset

Study	Method	Sensor location(s)	Windowing	Evaluation protocol	Accuracy (%)
Proposed	XGBoost + HPO for model size-constrained	4 IMUs (AS+BS, both sides)	6 seconds, 50% overlap	3-scenario split	93.21
[24]	1D-CNN (VGG-based)	Not specified	Not specified	80/20 split	99.3
[23]	Transformer	Not specified	Not specified	80/20 split + test on PVS 9	97.0 ($F1$), 80.41 (PVS 9)
[20]	XGBoost + HFW feature selection	3 IMUs (DB, left wheel, right wheel)	5 seconds, 0% overlap	70/30 split	94.2
[9]	SepRNet-1D	2 IMUs (DB, both sides)	3 seconds, 0% overlap	3-scenario split	94.69
[8]	CNN-LSTM (FFT + Stacked DWT)	2 IMUs (DB, both sides)	3 seconds, 0% overlap	66/34 split (Exp. 3 only)	94.78
[6]	ResBiGRU-SE	2 IMUs (AS+BS, left or right side only)	2 seconds, 50% overlap	5-fold CV	98.41
[5]	SE-ResNet	2 IMUs (AS+BS, left or right side only)	2 seconds, 50% overlap	5-fold CV	98.41
Menegazzo and Wangenheim (2021) [4]	CNN	2 IMUs (BS, both sides)	3 seconds, 0% overlap	3-scenario split	93.17 (Baseline)

The performance gap between the proposed method and modern DL architectures must be interpreted within the context of deployment hardware. While SepRNet-1D [9] was evaluated on a desktop CPU with 13 GB RAM, the current pipeline targets MCU-class hardware. Reliance on high-performance

computing platforms for model inference introduces significant financial and energy burdens [22]. A Cortex-M4 class MCU costs approximately 5-10 USD and can operate on low-power sources, enabling affordable and sustainable artificial intelligence inference [22]. In this context, achieving 93.21% accuracy on the PVS dataset – and 92.45% without any re-tuning – indicates that memory-efficient classical models remain a practical and competitive alternative to lightweight neural architectures.

4.4. Evaluation of GPS interpolation and pothole detection. The spatial accuracy of the proposed framework is evaluated by comparing detected coordinates against a static ground truth. This reference was established by recording GPS coordinates while remaining stationary at each pothole location to obtain a stable position. During test runs at average speeds of 20, 30, and 50 km/h, the localization error is measured as the geodesic distance between this static ground truth and two distinct points: (i) the raw GPS coordinates and (ii) the nearest interpolated coordinates assigned at the moment of impact. This procedure provides a quantitative measure of how closely the reconstructed trajectory aligns with the physical pothole position compared to raw satellite logs.

Table 14 summarizes the error metrics, including the mean error and root-mean-square error (RMSE). At 20 km/h, the raw GPS exhibits a mean error of 5.571 m, which is reduced to 1.835 m through interpolation. This improvement becomes more substantial as the vehicle speed increases and the raw GPS resolution gap widens. Specifically, the mean error decreases from 8.656 m to 2.833 m at 30 km/h, and from 11.893 m to 3.563 m at 50 km/h. Across all evaluated speeds, the proposed method significantly lowers the RMSE, notably decreasing from 11.911 m to 3.571 m at the 50 km/h threshold. These results indicate that the proposed interpolation method reduces the spatial gaps inherent in low-frequency updates, thereby improving the coordinate estimation as travel velocities increase.

Table 14. Comparison of pothole localization errors between raw GPS and the proposed velocity-aided interpolation across various average speeds

Average speed	**20 km/h**		**30 km/h**		**50 km/h**	
Indexes	**Normal**	**Proposed**	**Normal**	**Proposed**	**Normal**	**Proposed**
Min	4.849	1.474	7.841	2.209	11.014	3.278
Max	5.856	2.222	9.924	3.884	13.074	4.090
Mean	5.571	1.835	8.656	2.833	11.893	3.563
SD	0.299	0.220	0.779	0.572	0.679	0.233
RMSE	5.578	1.847	8.687	2.884	11.911	3.571

Normal: without interpolation; Proposed: interpolation method was applied

4.5. Inference latency and power consumption. The suitability of the optimized XGBoost model for resource-constrained hardware is validated through a computational complexity analysis on the ESP32 MCU (80 MHz). During operation, the system acquires 4200 samples within a 6-second window from seven sensor channels at a sampling frequency of 100 Hz. The exported C++ header file, occupying 975 kB of Flash memory, contains 126 decision trees with a maximum depth of 6. Analytical profiling of the 22951 instruction lines shows that the execution of a single prediction requires 16632 clock cycles, resulting in an inference latency of approximately 0.208 ms. This latency represents a significant reduction compared to the 2.64 ms and 3.47 ms reported for the SepRNet-1D and SepSERNet-1D architectures, respectively [9]. Furthermore, the end-to-end processing pipeline, encompassing feature extraction and model inference, incurs a total latency of 25.61 ms. This performance is significantly superior to the total processing time of 110.08 ms reported for the smartphone-based Asfault system [29]. Given the 3-second interval between recognition instances, the total processing time accounts for less than 1% of the available duty cycle, allowing for real-time operation while maintaining significant overhead for tasks such as GPS logging and data transmission.

Power consumption is evaluated based on the operational states of the hardware. The device maintains an idle power of 37-92.5 mW (10-25 mA), which increases to 314.5-388.5 mW (85-105 mA) during data acquisition from the sensors and GPS module. The inference phase requires 407-481 mW (110-130 mA), while LTE data transmission involves peaks between 666 and 814 mW (180-220 mA). Based on an average operating power of approximately 462.5 mW, the 2000 mAh battery provides about 13.6 hours of autonomous operation. This duration is sufficient to maintain system functionality during electrical instabilities until the vehicle power system is addressed. This performance is noteworthy when compared to smartphone-based applications, where the Asfault system demonstrated a battery drain of 13% per hour on a larger 2900 mAh battery, resulting in approximately 7.7 hours of operation [29].

The scalability of the proposed system for large-scale urban environments is supported by the efficiency of the edge-computing pipeline. By performing feature extraction and classification directly on the MCU, only high-level metadata – such as road surface types and coordinates – are transmitted to the central server. This significantly reduces network bandwidth requirements and server-side computational load, even in high-traffic scenarios with numerous active nodes. Furthermore, the low unit cost of the sensor node enables the deployment of extensive crowdsourcing fleets. While the current framework is validated on vehicles, there is significant potential for

extrapolation into multi-agent systems involving swarms of wheeled ground robots [30, 31]. In such configurations, autonomous agents and conventional vehicle fleets can collaborate through collective data fusion to refine spatial maps, ensuring high-resolution monitoring across broad geographical areas.

5. Conclusion. This research developed an IoT-based framework for RSC using a lightweight feature set and memory-constrained ML. The results show that optimized classical models achieve performance comparable to DL architectures while requiring significantly fewer computational resources. This efficiency allows the system to operate in real-time on low-cost MCU.

The framework also integrates GPS interpolation and pothole detection algorithms to improve trajectory reconstruction and hazard identification. Future work will focus on a real-time system to provide road-condition maps and automated alerts. This system will assist authorities in maintenance planning and provide road users with timely information to improve travel safety.

Data usage statement. The complete dataset is available at: 10.21227/x6av-va13. Researchers who wish to access this dataset must cite this paper in their publications and contact hieu.daoto@phenikaa-uni.edu.vn for further information.

Source code. HieuSSALAB/Real-TimeRoadSurface

Reference.

1. Carlos M.R., Aragon M.E., Gonzalez L.C., Escalante H.J., Martınez F. Evaluation of detection approaches for road anomalies based on accelerometer readings – addressing who's who. IEEE Transactions on Intelligent Transportation Systems. 2018. vol. 19. no. 10. pp. 3334–3343. DOI: 10.1109/TITS.2017.2773084.
2. Ganguly B., Dey D., Munshi S. An unsupervised learning approach for road anomaly segmentation using RGB-D sensor for advanced driver assistance system. IEEE Transactions on Intelligent Transportation Systems. 2022. vol. 23. no. 10. pp. 19042–19053. DOI: 10.1109/TITS.2022.3164847.
3. Duong C.C., Nguyen T.T., Duong V.T., Tran D.-N., Chinh T.M., Le A.N., Tran D.-T. Smartphone-based sensing for intelligent inland waterway transportation. International Journal of Interactive Mobile Technologies (iJIM). 2020. vol. 14. no. 18. pp. 195–203. DOI: 10.3991/ijim.v14i18.16449.
4. Menegazzo J., von Wangenheim A. Road surface type classification based on inertial sensors and machine learning: A comparison between classical and deep machine learning approaches for multi-contextual real-world scenarios. Computing. 2021. vol. 103. no. 10. pp. 2143–2170. DOI: 10.1007/s00607-021-00914-0.
5. Hnoohom N., Mekruksavanich S., Jitpattanakul A. A comprehensive evaluation of state-of-the-art deep learning models for road surface type classification. Intelligent Automation & Soft Computing. 2023. vol. 37. no. 2. pp. 1275–1291. DOI: 10.32604/i-asc.2023.038584.
6. Mekruksavanich S., Rojanavasu P., Srisungsittisunti B., Plengvittaya C., Phaphan W., Jitpattanakul A. Enhancing intelligent transportation systems: A deep learning approach

for terrain recognition using vehicular inertial sensors. Lobachevskii Journal of Mathematics. 2024. vol. 45. no. 12. pp. 6324–6342. DOI: 10.1134/S1995080224607628.
7. Sattar S., Li S., Chapman M. Developing a near real-time road surface anomaly detection approach for road surface monitoring. Measurement. 2021. vol. 185.
8. Raslan E., Alrahmawy M.F., Mohammed Y., Tolba A. Evaluation of data representation techniques for vibration based road surface condition classification. Scientific Reports. 2024. vol. 14. no. 1. DOI: 10.1038/s41598-024-61757-1.
9. Manoni L., Orcioni S., Conti M. A lightweight 1D-CNN architecture for accurate and efficient road type classification using vibrational signals. IEEE Access. 2025. vol. 13. pp. 174349–174367. DOI: 10.1109/ACCESS.2025.3617943.
10. Botezatu A.-P., Burlacu A., Orhei C. A review of deep learning advancements in road analysis for autonomous driving. Applied Sciences. 2024. vol. 14. no. 11. DOI: 10.3390/app14114705.
11. Yigit H., Koylu H., Eken S. Estimation of road surface type from brake pressure pulses of ABS. Expert Systems with Applications. 2023. vol. 212. DOI: 10.1016/j.eswa.2022.118726.
12. Biju S., Chammam A., Askar S., Rodrigues P., Jalalnezhad M. Prediction-based controller radial neural network for the traction control system. Journal of Vibration and Control. 2024. DOI: 10.1177/10775463241296911.
13. Martinez-Rıos E.A., Bustamante-Bello M.R., Arce-Saenz L.A. A review of road surface anomaly detection and classification systems based on vibration-based techniques. Applied Sciences. 2022. vol. 12. no. 19. DOI: 10.3390/app12199413.
14. Kim Y.-M., Kim Y.-G., Son S.-Y., Lim S.-Y., Choi B.-Y., Choi D.-H. Review of recent automated pothole-detection methods. Applied Sciences. 2022. vol. 12. no. 11. DOI: 10.3390/app12115320.
15. Manoni L., Orcioni S., Conti M. Recent advancements in deep learning techniques for road condition monitoring: A comprehensive review. IEEE Access. 2024. vol. 12. pp. 154271–154293. DOI: 10.1109/ACCESS.2024.3481649.
16. Coenen T.B., Golroo A. A review on automated pavement distress detection methods. Cogent Engineering. 2017. vol. 4. no. 1. DOI: 10.1080/23311916.2017.1374822.
17. Cao W., Liu Q., He Z. Review of pavement defect detection methods. IEEE Access. 2020. vol. 8. pp. 14531–14544. DOI: 10.1109/ACCESS.2020.2966881.
18. Dib J., Sirlantzis K., Howells G. A review on negative road anomaly detection methods. IEEE Access. 2020. vol. 8. pp. 57298–57316. DOI: 10.1109/ACCESS.2020.2982220.
19. Peraka N.S.P., Biligiri K.P. Pavement asset management systems and technologies: A review. Automation in Construction. 2020. vol. 119. DOI: 10.1016/j.autcon.2020.103336.
20. Cong N.V., Tran D.-N., Long T.T., Thao N.G.M., Tran D.-T. Hybrid feature selection for real-time road surface classification on low-end hardware: A machine learning approach. Results in Engineering. 2025. vol. 27. DOI: 10.1016/j.rineng.2025.105693.
21. Dao T.-H., Tran D.-N., Bui V.-H., Nguyen V.S., Hoa D.K., Thanh P.V., Tran D.-T. RFAR: A real-time firefighter activity recognition system using wearable accelerometer. IEEE Sensors Journal. 2025. vol. 25. no. 17. pp. 33674–33691. DOI: 10.1109/JSEN.2025.3593466.
22. Saha S.S., Sandha S.S., Srivastava M. Machine learning for microcontroller-class hardware: A review. IEEE Sensors Journal. 2022. vol. 22. no. 22. pp. 21362–21390. DOI: 10.1109/JSEN.2022.3210773.
23. Aslam I., Mahfuz S. Transformer-based classification of road conditions using vehicular sensor data. Procedia Computer Science. 2025. vol. 257. pp. 444–451. DOI: 10.1016/j.procs.2025.03.058.

24. Cui J., Zhang H., Wang X., Jing Y., Chou X. Research on road surface recognition algorithm based on vehicle vibration data. Sensors. 2025. vol. 25. no. 18. DOI: 10.3390/s25185642.
25. Thang N.V., Thang P.D., Kien L.M., Thu N.T., Dao T.-H. Development of low-cost road surface classification system using acceleration sensors on motorcycles. Edge Artificial Intelligence: Foundations, Techniques, and Applications. 2025. pp. 567–578. DOI: 10.1002/9781394355037.ch25.
26. Lu D.-N., Nguyen D.-N., Nguyen T.-H., Nguyen H.-N. Vehicle mode and driving activity detection based on analyzing sensor data of smartphones. Sensors. 2018. vol. 18. no. 4. DOI: 10.3390/s18041036.
27. Durap A. A comparative analysis of machine learning algorithms for predicting wave runup. Anthropocene Coasts. 2023. vol. 6. no. 1. DOI: 10.1007/s44218-023-00033-7.
28. Thomas N.S., Kaliraj S. An improved and optimized random forest based approach to predict the software faults. SN Computer Science. 2024. vol. 5. no. 5. DOI: 10.1007/s42979-024-02764-x.
29. Souza V.M., Giusti R., Batista A.J. Asfault: A low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning. Pervasive and Mobile Computing. 2018. vol. 51. pp. 121–137. DOI: 10.1016/j.pmcj.2018.10.008.
30. Khaleghian S., Taheri S. Terrain classification using intelligent tire. Journal of Terramechanics. 2017. vol. 71. pp. 15–24. DOI: 10.1016/j.jterra.2017.01.005.
31. Sebastian B., Ben-Tzvi P. Support vector machine based real-time terrain estimation for tracked robots. Mechatronics. 2019. vol. 62. DOI: 10.1016/j.mechatronics.2019.102260.

Viet-Hoan Bui — Student, Phenikaa School of Engineering, Phenikaa University. Research interests: integrated sensors, machine learning, advanced signal processing for healthcare applications. The number of publications — 4. 21011096@st.phenikaa-uni.edu.vn; Duongnoi, 12116, Hanoi, Vietnam; office phone: +84(869)891-098.

Duc-Nghia Tran — Ph.D., Researcher, Institute of Information Technology, Vietnam Academy of Science and Technology. Research interests: AI, IoT, electron paramagnetic resonance, parameter estimation, data analysis. The number of publications — 51. nghiatd@ioit.ac.vn; 18, Hoang Quoc, 10000, Hanoi, Vietnam; office phone: +84(936)866-335.

To-Hieu Dao — Ph.D., Lecturer, Phenikaa School of Engineering, Phenikaa University. Research interests: machine learning, indoor localization, IMU sensor fusion, signal processing, vibration-based sensing, embedded systems, real-time sensing applications. The number of publications — 12. hieu.daoto@phenikaa-uni.edu.vn; Duongnoi, 12116, Hanoi, Vietnam; office phone: +84(389)959-524.

Hoang Quang Trung — Ph.D., Lecturer, Phenikaa School of Engineering, Phenikaa University. Research interests: signal and data processing for telecommunication systems, internet of things (IoT). The number of publications — 10. trung.hoangquang@phenikaa-uni.edu.vn; Duongnoi, 12116, Hanoi, Vietnam; office phone: +84(389)959-524.

Pham Vu Kien — Student, Phenikaa School of Engineering, Phenikaa University. Research interests: IoT and intelligent sensing solutions for smart infrastructure and transportation systems. 20010645@st.phenikaa-uni.edu.vn; Duongnoi, 12116, Hanoi, Vietnam; office phone: +84(389)959-524.

Nguyen Van Thang — Ph.D., Lecturer, VNU University of Engineering and Technology. Research interests: IoT and sensor applications, biomedical electronics, signal processing. The number of publications — 20. nvthangdtvt@vnu.edu.vn; 144, Xuan Thuy, Hanoi, Vietnam; office phone: +84(389)959-524.

Duc-Tan Tran — Ph.D., Dr.Sci., Professor, Lecturer, Vice Dean of the Faculty, Phenikaa School of Engineering, Phenikaa University. Research interests: representation, processing,

analysis, and communication of information embedded in signals and datasets. The number of publications — 150. tan.tranduc@phenikaa-uni.edu.vn; Duongnoi, 12116, Hanoi, Vietnam; office phone: +84(904)182-389.

Acknowledgements. This research is funded by Phenikaa University, number PU2024-1-A-01. Thanks to the Institute of Information Technology (IoIT-VAST) for supporting this research and allowing us to use the «IoT and Robot intensive laboratory» equipment.

УДК 006.72 **DOI** 10.15622/ia.25.2.7

В.-Х. Буи , Д.-Н. Тран, Т.-Х. Дао, Х.-К. Чунг, Ф.-В. Киен, Н.-В. Тханг, Д.-Т. Тран

IDRRS: ИНЕРЦИАЛЬНОЕ IOT-УСТРОЙСТВО ДЛЯ КЛАССИФИКАЦИИ ДОРОЖНОГО ПОКРЫТИЯ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ И ПОВЫШЕНИЯ ТОЧНОСТИ ОЦЕНКИ МЕСТОПОЛОЖЕНИЯ

Буй В.-Х., Тран Д.-Н., Дао Т.-Х., Чунг Х.К., Киен Ф.В., Тханг Н.В., Тран Д.-Т. **IDRRS: инерциальное IoT-устройство для классификации дорожного покрытия в режиме реального времени и повышения точности оценки местоположения.**

Аннотация. Мониторинг состояния дорожного покрытия является важной задачей, направленной на повышение безопасности дорожного движения и оптимизацию обслуживания транспортной инфраструктуры. В настоящей работе разработана инерциальная IoT-система, предназначенная для классификации типов дорожного покрытия в режиме реального времени, обнаружения выбоин и повышения точности оценки местоположения транспортного средства. Предложенная система включает модель XGBoost, адаптированную для развертывания на микроконтроллерах с ограниченным объемом памяти, алгоритм интерполяции GPS с использованием данных о скорости движения, а также метод обнаружения выбоин на основе индекса аномалий. Экспериментальная оценка проведена на собственном наборе данных и общедоступном наборе данных PVS. Достигнутая точность классификации составила 95.39% и 93.21% соответственно. Для анализа переносимости модель, обученная на собственном наборе данных, была применена к набору PVS без дополнительного обучения и обеспечила точность 92.45%, что подтверждает устойчивость предложенного подхода к смене источника данных. Применение процедуры интерполяции GPS позволило снизить среднюю ошибку локализации с 5.571-11.893 м до 1.835-3.563 м при скоростях движения транспортного средства от 20 до 50 км/ч. Дополнительным вкладом работы является публикация собственного набора данных, содержащего вибрационные сигнатуры типовых дорожных покрытий, что способствует дальнейшему развитию методов интеллектуальной классификации дорожного покрытия.

Ключевые слова: инерциальный датчик, режим реального времени, микроконтроллер, машинное обучение, интернет вещей.

Литература

1. Carlos M.R., Aragon M.E., Gonzalez L.C., Escalante H.J., Martınez F. Evaluation of detection approaches for road anomalies based on accelerometer readings – addressing who's who. IEEE Transactions on Intelligent Transportation Systems. 2018. vol. 19. no. 10. pp. 3334–3343. DOI: 10.1109/TITS.2017.2773084.
2. Ganguly B., Dey D., Munshi S. An unsupervised learning approach for road anomaly segmentation using RGB-D sensor for advanced driver assistance system. IEEE Transactions on Intelligent Transportation Systems. 2022. vol. 23. no. 10. pp. 19042–19053. DOI: 10.1109/TITS.2022.3164847.
3. Duong C.C., Nguyen T.T., Duong V.T., Tran D.-N., Chinh T.M., Le A.N., Tran D.-T. Smartphone-based sensing for intelligent inland waterway transportation. International Journal of Interactive Mobile Technologies (iJIM). 2020. vol. 14. no. 18. pp. 195–203. DOI: 10.3991/ijim.v14i18.16449.

4. Menegazzo J., von Wangenheim A. Road surface type classification based on inertial sensors and machine learning: A comparison between classical and deep machine learning approaches for multi-contextual real-world scenarios. Computing. 2021. vol. 103. no. 10. pp. 2143–2170. DOI: 10.1007/s00607-021-00914-0.
5. Hnoohom N., Mekruksavanich S., Jitpattanakul A. A comprehensive evaluation of state-of-the-art deep learning models for road surface type classification. Intelligent Automation & Soft Computing. 2023. vol. 37. no. 2. pp. 1275–1291. DOI: 10.32604/i-asc.2023.038584.
6. Mekruksavanich S., Rojanavasu P., Srisungsittisunti B., Plengvittaya C., Phaphan W., Jitpattanakul A. Enhancing intelligent transportation systems: A deep learning approach for terrain recognition using vehicular inertial sensors. Lobachevskii Journal of Mathematics. 2024. vol. 45. no. 12. pp. 6324–6342. DOI: 10.1134/S1995080224607628.
7. Sattar S., Li S., Chapman M. Developing a near real-time road surface anomaly detection approach for road surface monitoring. Measurement. 2021. vol. 185.
8. Raslan E., Alrahmawy M.F., Mohammed Y., Tolba A. Evaluation of data representation techniques for vibration based road surface condition classification. Scientific Reports. 2024. vol. 14. no. 1. DOI: 10.1038/s41598-024-61757-1.
9. Manoni L., Orcioni S., Conti M. A lightweight 1D-CNN architecture for accurate and efficient road type classification using vibrational signals. IEEE Access. 2025. vol. 13. pp. 174349–174367. DOI: 10.1109/ACCESS.2025.3617943.
10. Botezatu A.-P., Burlacu A., Orhei C. A review of deep learning advancements in road analysis for autonomous driving. Applied Sciences. 2024. vol. 14. no. 11. DOI: 10.3390/app14114705.
11. Yigit H., Koylu H., Eken S. Estimation of road surface type from brake pressure pulses of ABS. Expert Systems with Applications. 2023. vol. 212. DOI: 10.1016/j.eswa.2022.118726.
12. Biju S., Chammam A., Askar S., Rodrigues P., Jalalnezhad M. Prediction-based controller radial neural network for the traction control system. Journal of Vibration and Control. 2024. DOI: 10.1177/10775463241296911.
13. Martinez-Rıos E.A., Bustamante-Bello M.R., Arce-Saenz L.A. A review of road surface anomaly detection and classification systems based on vibration-based techniques. Applied Sciences. 2022. vol. 12. no. 19. DOI: 10.3390/app12199413.
14. Kim Y.-M., Kim Y.-G., Son S.-Y., Lim S.-Y., Choi B.-Y., Choi D.-H. Review of recent automated pothole-detection methods. Applied Sciences. 2022. vol. 12. no. 11. DOI: 10.3390/app12115320.
15. Manoni L., Orcioni S., Conti M. Recent advancements in deep learning techniques for road condition monitoring: A comprehensive review. IEEE Access. 2024. vol. 12. pp. 154271–154293. DOI: 10.1109/ACCESS.2024.3481649.
16. Coenen T.B., Golroo A. A review on automated pavement distress detection methods. Cogent Engineering. 2017. vol. 4. no. 1. DOI: 10.1080/23311916.2017.1374822.
17. Cao W., Liu Q., He Z. Review of pavement defect detection methods. IEEE Access. 2020. vol. 8. pp. 14531–14544. DOI: 10.1109/ACCESS.2020.2966881.
18. Dib J., Sirlantzis K., Howells G. A review on negative road anomaly detection methods. IEEE Access. 2020. vol. 8. pp. 57298–57316. DOI: 10.1109/ACCESS.2020.2982220.
19. Peraka N.S.P., Biligiri K.P. Pavement asset management systems and technologies: A review. Automation in Construction. 2020. vol. 119. DOI: 10.1016/j.autcon.2020.103336.
20. Cong N.V., Tran D.-N., Long T.T., Thao N.G.M., Tran D.-T. Hybrid feature selection for real-time road surface classification on low-end hardware: A machine learning approach. Results in Engineering. 2025. vol. 27. DOI: 10.1016/j.rineng.2025.105693.

21. Dao T.-H., Tran D.-N., Bui V.-H., Nguyen V.S., Hoa D.K., Thanh P.V., Tran D.-T. RFAR: A real-time firefighter activity recognition system using wearable accelerometer. IEEE Sensors Journal. 2025. vol. 25. no. 17. pp. 33674–33691. DOI: 10.1109/JSEN.2025.3593466.
22. Saha S.S., Sandha S.S., Srivastava M. Machine learning for microcontroller-class hardware: A review. IEEE Sensors Journal. 2022. vol. 22. no. 22. pp. 21362–21390. DOI: 10.1109/JSEN.2022.3210773.
23. Aslam I., Mahfuz S. Transformer-based classification of road conditions using vehicular sensor data. Procedia Computer Science. 2025. vol. 257. pp. 444–451. DOI: 10.1016/j.procs.2025.03.058.
24. Cui J., Zhang H., Wang X., Jing Y., Chou X. Research on road surface recognition algorithm based on vehicle vibration data. Sensors. 2025. vol. 25. no. 18. DOI: 10.3390/s25185642.
25. Thang N.V., Thang P.D., Kien L.M., Thu N.T., Dao T.-H. Development of low-cost road surface classification system using acceleration sensors on motorcycles. Edge Artificial Intelligence: Foundations, Techniques, and Applications. 2025. pp. 567–578. DOI: 10.1002/9781394355037.ch25.
26. Lu D.-N., Nguyen D.-N., Nguyen T.-H., Nguyen H.-N. Vehicle mode and driving activity detection based on analyzing sensor data of smartphones. Sensors. 2018. vol. 18. no. 4. DOI: 10.3390/s18041036.
27. Durap A. A comparative analysis of machine learning algorithms for predicting wave runup. Anthropocene Coasts. 2023. vol. 6. no. 1. DOI: 10.1007/s44218-023-00033-7.
28. Thomas N.S., Kaliraj S. An improved and optimized random forest based approach to predict the software faults. SN Computer Science. 2024. vol. 5. no. 5. DOI: 10.1007/s42979-024-02764-x.
29. Souza V.M., Giusti R., Batista A.J. Asfault: A low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning. Pervasive and Mobile Computing. 2018. vol. 51. pp. 121–137. DOI: 10.1016/j.pmcj.2018.10.008.
30. Khaleghian S., Taheri S. Terrain classification using intelligent tire. Journal of Terramechanics. 2017. vol. 71. pp. 15–24. DOI: 10.1016/j.jterra.2017.01.005.
31. Sebastian B., Ben-Tzvi P. Support vector machine based real-time terrain estimation for tracked robots. Mechatronics. 2019. vol. 62. DOI: 10.1016/j.mechatronics.2019.102260.

Буй Вьет-Хоан — студент, инженерная школа, Университет Феникаа. Область научных интересов: интегрированные сенсорные системы, методы машинного обучения, современные методы обработки сигналов для применения в сфере здравоохранения. Число научных публикаций — 4. 21011096@st.phenikaa-uni.edu.vn; Дуонгной, 12116, Ханой, Вьетнам; р.т.: +84(869)891-098.

Тран Дук-Нгиа — Ph.D., научный сотрудник, институт информационных технологий, Вьетнамская академия наук и технологий. Область научных интересов: искусственный интеллект, интернет вещей (IoT), электронный парамагнитный резонанс, оценивание параметров, анализ данных. Число научных публикаций — 51. nghiatd@ioit.ac.vn; Хоанг Куок, 18, 10000, Ханой, Вьетнам; р.т.: +84(936)866-335.

Дао Ту-Хиеу — Ph.D., преподаватель, инженерная школа, Университет Феникаа. Область научных интересов: машинное обучение, определение местоположения внутри помещений, объединение данных с инерциальных измерительных блоков (IMU), обработка сигналов, датчики на основе вибрации, встроенные системы, приложения для сбора данных в реальном времени. Число научных публикаций — 12. hieu.daoto@phenikaa-uni.edu.vn; Дуонгной, 12116, Ханой, Вьетнам; р.т.: +84(389)959-524.

Чунг Хоанг Куанг — Ph.D., преподаватель, инженерная школа, Университет Феникаа. Область научных интересов: обработка сигналов и данных для телекоммуникационных систем, интернет вещей (IoT). Число научных публикаций — 10. trung.hoangquang@phenikaa-uni.edu.vn; Дуонгной, 12116, Ханой, Вьетнам; р.т.: +84(389)959-524.

Киен Фам Ву — студент, инженерная школа, Университет Феникаа. Область научных интересов: IoT, интеллектуальные сенсорные системы, поддерживающие умную инфраструктуру, транспортные системы. 20010645@st.phenikaa-uni.edu.vn; Дуонгной, 12116, Ханой, Вьетнам; р.т.: +84(389)959-524.

Тханг Нгуен Ван — Ph.D., преподаватель, Университет инженерии и технологий Вьетнамского национального университета. Область научных интересов: приложения интернета вещей (IoT) и сенсорных систем, биомедицинская электроника, обработка сигналов. Число научных публикаций — 20. nvthangdtvt@vnu.edu.vn; Суан Туй, 144, Ханой, Вьетнам; р.т.: +84(389)959-524.

Тран Дук-Тан — Ph.D., Dr.Sci., профессор, преподаватель, заместитель декана факультета, инженерная школа, Университет Феникаа. Область научных интересов: представление, обработка, анализ и передача информации, содержащейся в сигналах и наборах данных. Число научных публикаций — 150. tan.tranduc@phenikaa-uni.edu.vn; Дуонгной, 12116, Ханой, Вьетнам; р.т.: +84(904)182-389.

Поддержка исследований. Работа выполнена при финансовой поддержке Университета Феникаа (проект № PU2024-1-A-01). Авторы выражают благодарность Институту информационных технологий (ИИТ ВАНТ) за поддержку исследования и предоставленную возможность использовать оборудование лаборатории «Интенсивные робототехнические и IoT-системы».

УДК 338.001.36 **DOI** 10.15622/ia.25.2.8

Р.С. Рогулин

ДВУХЭШЕЛОННАЯ МОДЕЛЬ ТРАНСПОРТНОЙ СИСТЕМЫ И МУРАВЬИНЫЙ АЛГОРИТМ: АНАЛИЗ МАСШТАБИРУЕМОСТИ ВЫЧИСЛИТЕЛЬНЫХ РЕШЕНИЙ

Рогулин Р.С. **Двухэшелонная модель транспортной системы и муравьиный алгоритм: анализ масштабируемости вычислительных решений.**

Аннотация. В статье рассматривается двухэшелонная модель транспортной системы, предназначенная для описания и анализа распределительных логистических процессов с промежуточными хабами и конечными потребителями. Модель учитывает совместную оптимизацию маршрутов магистрального уровня и маршрутов распределения при наличии ограничений по вместимости транспортных средств и требований по удовлетворению спроса. Рассматриваемая задача относится к классу NP-трудных, что существенно ограничивает применение точных методов оптимизации при росте размерности транспортного графа. Для решения предложенной модели используется двухэшелонная модификация алгоритма муравьиной колонии (2E-ACO), в которой процессы формирования решений для первого и второго эшелонов формализованы раздельно, но согласованы через единую целевую функцию, включающую транспортные затраты и штрафы за необслуженный спрос. Основное внимание в работе уделено вычислительному эксперименту, направленному на анализ масштабируемости и устойчивости алгоритма при увеличении мощности множества потребителей, числа хабов и сложности транспортной инфраструктуры. Эксперименты проводятся в режиме масштабируемых ресурсов, что позволяет отделить влияние алгоритмических решений от эффектов ресурсных ограничений. Для оценки воспроизводимости используются многократные независимые запуски с фиксированными силами генератора случайных чисел. Полученные результаты демонстрируют предсказуемый рост вычислительных затрат при увеличении размерности модели и устойчивость качества решений. Сравнение с намеренно простой базовой жадной эвристикой, используемой в качестве нижней оценки качества решений, показывает, что алгоритм 2E-ACO обеспечивает сопоставимый уровень обслуживания спроса при более высоких вычислительных затратах, обусловленных итерационным характером поиска. Представленные результаты подтверждают применимость предложенной модели и алгоритма для исследования крупномасштабных двухэшелонных транспортных систем.

Ключевые слова: двухэшелонная транспортная задача, муравьиный алгоритм, метаэвристики, масштабируемость, вычислительный эксперимент, логистические системы.

1. Введение.

Современные логистические системы характеризуются высокой пространственной распределённостью объектов, многоуровневой структурой потоков и существенной комбинаторной сложностью задач планирования перевозок. В таких условиях особую актуальность приобретают двухэшелонные транспортные модели, в которых процессы магистральной доставки от депо к промежуточным распределительным пунктам и последующего распределения продукции к конечным потребителям рассматриваются

как взаимосвязанные, но структурно различающиеся этапы единой системы.

Формально двухэшелонная транспортная задача предполагает наличие центрального депо I_0, множества промежуточных хабов I_1, множества потребителей I_2 и транспортной инфраструктуры, представляемой в виде ориентированного или неориентированного взвешенного графа, включающего также транзитные вершины I_T. Первый эшелон ориентирован на построение маршрутов между элементами множества $I_0 \cup I_1$ и по своей структуре близок к задаче коммивояжёра или задаче маршрутизации транспортных средств. Второй эшелон связан с распределением потоков от хабов I_1 к потребителям I_2 при наличии ограничений по вместимости транспортных средств и объёму спроса.

Большинство существующих подходов к решению двухэшелонных задач маршрутизации можно условно разделить на две группы. К первой группе относятся точные методы, основанные на декомпозиции, генерации колонок и процедурах ветвей и границ (branch-and-price, branch-price-and-cut). Эти методы позволяют получать оптимальные решения или строгие оценки оптимальности, однако их вычислительная трудоёмкость резко возрастает при увеличении мощности множеств I_1 и I_2, что существенно ограничивает их применимость для задач большой размерности. Вторая группа представлена эвристическими и метаэвристическими методами, среди которых наибольшее распространение получили подходы семейства Adaptive Large Neighborhood Search (ALNS), а также гибридные алгоритмы, ориентированные на поиск высококачественных приближённых решений в приемлемое время.

Отдельный класс работ посвящён применению муравьиных алгоритмов (Ant Colony Optimization, ACO) к задачам маршрутизации. Муравьиные алгоритмы зарекомендовали себя как эффективный инструмент решения комбинаторных оптимизационных задач благодаря сочетанию стохастического глобального поиска и накопления эвристической информации. Вместе с тем большинство существующих ACO-подходов ориентированы либо на классические одноуровневые задачи маршрутизации, либо на последовательную оптимизацию отдельных эшелонов в многоуровневых постановках. Вопрос согласованной оптимизации маршрутов разных эшелонов в рамках единого муравьиного алгоритма, использующего общий механизм оценки качества решений, остаётся недостаточно исследованным.

Дополнительную сложность представляет учёт реальной транспортной инфраструктуры, в которой маршруты между значимыми узлами сети (депо, хабы, потребители) проходят через систему транзитных вершин I_T. Во многих работах транспортная сеть аппроксимируется полным графом либо транзитные вершины учитываются неявно через заранее вычисленные расстояния. Такое допущение упрощает вычисления, но снижает адекватность модели для анализа сложных пространственно распределённых логистических систем.

В данной работе предлагается двухэшелонная транспортная модель, в которой структура транспортной сети явно представляется в виде графа с транзитными вершинами, а также модифицированный муравьиный алгоритм (2E-ACO), обеспечивающий согласованную оптимизацию маршрутов первого и второго эшелонов. В отличие от последовательных и декомпозиционных подходов, предлагаемый алгоритм использует единый критерий качества решения и интегрирует графовые методы вычисления кратчайших путей с эвристическим поиском в пространстве комбинаторных маршрутов.

Основной вклад работы заключается в разработке алгоритмического подхода к решению двухэшелонной транспортной задачи на графах с транзитными вершинами, а также в проведении систематического вычислительного эксперимента, направленного на анализ масштабируемости, устойчивости и вычислительной эффективности предложенного алгоритма при росте мощности множеств I_1, I_2 и I_T. Результаты экспериментов позволяют оценить применимость метода для задач большой размерности и сопоставить его поведение с базовыми эвристическими подходами.

2. Формулировка задачи. Рассматривается двухэшелонная транспортная задача на взвешенном графе, описывающем структуру логистической сети с транзитными вершинами. Транспортная сеть представляется в виде графа

$$G = (V, E),$$

где множество вершин имеет вид

$$V = I_0 \cup I_1 \cup I_2 \cup I_T.$$

Здесь I_0 – центральное депо, I_1 – множество промежуточных распределительных пунктов (хабов), I_2 – множество конечных потребителей, I_T – множество транзитных вершин, не являющихся

источниками или потребителями потоков. Каждой дуге $(i,j) \in E$ сопоставлен неотрицательный вес c_{ij}, характеризующий затраты или расстояние перемещения между вершинами i и j.

Транспортный процесс организован в два эшелона. Первый эшелон соответствует доставке грузов от депо I_0 к хабам I_1, второй эшелон – распределению грузов от хабов I_1 к потребителям I_2. Перемещение между элементами множеств I_0, I_1 и I_2 осуществляется по путям в графе G, которые могут проходить через транзитные вершины I_T.

2.1. Переменные и ограничения. Для первого эшелона требуется определить маршрут (или совокупность маршрутов), начинающийся и заканчивающийся в депо I_0 и проходящий через все вершины множества I_1. Данный маршрут определяет порядок обслуживания хабов и, в общем случае, эквивалентен задаче маршрутизации или задаче коммивояжёра на подграфе, индуцированном вершинами $I_0 \cup I_1$.

Для второго эшелона для каждого хаба $i \in I_1$ формируется набор маршрутов доставки к потребителям $j \in I_2$, закреплённым за данным хабом. Каждый маршрут второго эшелона начинается и заканчивается в соответствующем хабе и обслуживается транспортным средством с ограниченной вместимостью Q. Спрос каждого потребителя $j \in I_2$ задаётся величиной $d_j \geq 0$ и должен быть полностью либо частично удовлетворён в рамках доступных ресурсов второго эшелона.

Для маршрутов второго эшелона выполняются следующие ограничения:

– суммарный обслуживаемый спрос на каждом маршруте не превышает вместимость транспортного средства Q;

– каждый потребитель может обслуживаться не более одного раза;

– маршруты строятся по допустимым путям в графе G.

2.2. Целевая функция. Качество решения двухэшелонной транспортной задачи оценивается с использованием единого целевого функционала

$$F = F_1 + F_2 + F_p,$$

где:

– F_1 – суммарные затраты на маршруты первого эшелона, включающие перемещения между депо I_0 и хабами I_1;

– F_2 – суммарные затраты на маршруты второго эшелона между хабами I_1 и потребителями I_2;

– F_p – штрафная составляющая, отражающая неудовлетворённый спрос потребителей либо нарушения ограничений.

Компоненты F_1 и F_2 вычисляются на основе кратчайших путей в графе G между соответствующими вершинами, что позволяет явно учитывать структуру транспортной инфраструктуры и наличие транзитных вершин I_T.

2.3. Класс сложности задачи. Рассматриваемая двухэшелонная транспортная задача относится к классу NP-трудных задач, поскольку включает в себя в качестве частных случаев задачу коммивояжёра и задачу маршрутизации транспортных средств с ограничениями по вместимости. Наличие двух взаимосвязанных эшелонов и явное представление транспортной сети в виде графа с транзитными вершинами дополнительно увеличивают размерность и комбинаторную сложность задачи.

В связи с этим применение точных методов оптимизации для задач большой размерности существенно ограничено вычислительными затратами, что обосновывает использование эвристических и метаэвристических алгоритмов, ориентированных на поиск высококачественных приближённых решений в разумное время.

3. Обзор литературы

3.1. Двухэшелонные модели транспортных систем: постановки и варианты ограничений. Двухэшелонные модели распределения формализуют транспортную систему, в которой перемещение грузов осуществляется на двух взаимосвязанных уровнях: магистральном (первый эшелон) и распределительном (второй эшелон). Типичная интерпретация соответствует схемам городской логистики и доставки последней мили, где грузы консолидируются на промежуточных объектах (хабах, сателлитах), после чего доставляются конечным потребителям. Систематизация исследований по 2E-VRP представлена в обзорной работе [1], где предложена классификация постановок по структуре сети, типам промежуточных объектов, вариантам синхронизации эшелонов и набору операционных ограничений.

В рамках 2E-VRP выделяются постановки с временными окнами и межэшелонной синхронизацией, что приводит к появлению дополнительных ограничений на расписания и усложняет структуру допустимости решений [2]. Современные расширения включают 2E-VRP с одновременными заборами, доставками и дедлайнами [3], модели с промежуточными объектами различного назначения и привлечением нерегулярных водителей [4], а также задачи для оптовых рынков сельскохозяйственной продукции с балансировкой нагрузки

[5]. Недавняя работа [6] рассматривает двухэшелонную задачу с синхронизацией для интегрированной водно-наземной транспортной системы, что демонстрирует расширение методологии на мультимодальные сети.

Ключевая особенность двухэшелонных постановок состоит в комбинаторной связи между эшелонами: решения первого уровня (обход хабов, распределение потоков) определяют допустимость и стоимость решений второго уровня. Поэтому оптимизация ведётся по интегральному критерию, агрегирующему вклад обоих эшелонов и штрафы за невыполнение ограничений [1].

3.2. Математические формулировки и точные методы. Для двухэшелонных задач характерно использование смешанно-целочисленных формулировок (MILP), включающих маршрутизационные переменные на двух уровнях, ограничения баланса потоков и связующие ограничения согласованности эшелонов. Базовые математические модели и матэвристики для 2E-CVRP представлены в фундаментальной работе [7], которая остаётся методологически значимой как источник структур ограничений и принципов декомпозиции.

Для постановок с временными окнами и синхронизацией применяются точные схемы branch-price-and-cut (BPC), в которых маршруты второго эшелона порождаются в задаче ценообразования (ESPPRC/RCSP), а глобальная оптимальность достигается за счёт ветвления и отсечения [2]. Такие методы обеспечивают оптимальные решения для инстансов умеренной размерности, но демонстрируют типичное ограничение: резкий рост вычислительной сложности при увеличении числа потребителей и промежуточных объектов.

Современное развитие точных методов включает декомпозиции Бендерса для задач производства и распределения в условиях неопределённости [8, 9], а также подходы к учёту риск-компонент и построению устойчивых решений при вариативности параметров [10, 11]. Работа [12] предлагает двухстадийную робастную оптимизацию для проектирования логистических сетей с использованием эволюционных вычислений.

3.3. Метаэвристики и матэвристики для маршрутизации. Поскольку двухэшелонные постановки относятся к NP-трудным задачам и содержат элементы TSP-подзадач (первый эшелон) и VRP-подзадач с ресурсными ограничениями (второй эшелон), практико-ориентированные решения строятся на базе метаэвристик и матэвристик [13]. Доминирующее положение занимает семейство

алгоритмов Adaptive Large Neighborhood Search (ALNS), успешно применяемых к 2E-VRP и его расширениям [14 – 17].

ALNS обеспечивает платформу для построения гибридных решателей за счёт набора операторов разрушения-восстановления и адаптивного выбора операторов. Систематический обзор ALNS-операторов для VRP представлен в работе [18], которая задаёт рамку для сравнения с альтернативными метаэвристиками и позволяет корректно позиционировать ACO-подход относительно доминирующих в практике ALNS-решателей.

Работа [19] предлагает матэвристику для 2E-VRP с многократными рейсами, смешанным спросом и временными окнами, интегрирующую точную формулировку для первого эшелона в ALNS-фреймворк для второго эшелона. В применениях к задачам с транзитными узлами и мобильными сателлитами распространён подход, при котором глобальная структура решения строится конструктивно, затем усиливается локальным поиском, а межэшелонные связи поддерживаются через общую функцию качества [20, 21].

3.4. Алгоритмы муравьиной колонии в задачах маршрутизации. Алгоритмы муравьиной колонии (ACO) относятся к ключевым классам роевых метаэвристик; их сильной стороной является наличие явной памяти поиска (феромон), обеспечивающей баланс между интенсификацией и диверсификацией [22]. В современных работах ACO модифицируется для усложнённых VRP-постановок: показаны улучшения для задач со split pickup/split delivery [23], а также гибридные варианты ACO для мультикомпартментных постановок маршрутизации [24].

Существенная тенденция состоит в том, что ACO редко используется изолированно: как правило, он комбинируется с локальным улучшением или problem-specific процедурами построения допустимых решений [1, 25]. Это особенно значимо для двухэшелонных моделей, где необходимо согласовывать решения на двух уровнях. Бенчмарки для задач маршрутизации электрических транспортных средств представлены в работе [26].

Вместе с тем вопрос согласованной оптимизации обоих эшелонов в рамках единого ACO-механизма с общей функцией качества остаётся недостаточно изученным. Большинство существующих подходов используют либо последовательную оптимизацию эшелонов, либо декомпозиционные схемы, что затрудняет анализ интегрального поведения системы.

3.5. Эволюционные и гибридные метаэвристики. Наряду с ACO и ALNS, для задач маршрутизации применяются эволюционные алгоритмы и рой частиц (GA/PSO), включая гибридные схемы для сложных ограничений [27 – 29]. В многоцелевых постановках развиваются гибридные GA/PSO-подходы, позволяющие учитывать конкурирующие критерии стоимости, времени и устойчивости [29].

Классические техники tabu search и simulated annealing сохраняют значение как компоненты гибридных решателей и как базовые механизмы избегания локальных минимумов [30 – 33]. При этом их роль в современных исследованиях чаще состоит в интеграции как модулей локального поиска, а не в самостоятельной конкуренции с ALNS и ACO.

3.6. Проектирование сети и контекст реальных транспортных систем. Двухэшелонность связана с задачами проектирования транспортных сетей и многостадийной оптимизацией. Развиваются подходы к проектированию сетей в условиях неопределённости, включая распределительно робастные двухстадийные модели для гуманитарных и логистических сетей [11], а также двухуровневые постановки в виде игр Штакельберга [34]. Обзор методов оптимизации маршрутов последней мили представлен в работах [35, 36], где систематизированы постановки и перспективы исследований.

3.7. Масштабируемость алгоритмов и экспериментальная методология. Анализ вычислительных свойств алгоритмов при росте размерности задачи является неотъемлемой частью исследований в области комбинаторной оптимизации. Применимы как классические подходы к оценке масштабируемости [37], так и современные методологии instance space analysis, связывающие характеристики инстансов с поведением эвристик [38].

Последний подход позволяет объяснить ситуации, когда различные процедуры дают близкие значения целевой функции на части инстансов: это может быть следствием структуры задачи и сжатия пространства улучшений при определённой параметризации. Стандартные тестовые наборы для маршрутизации [39] остаются важной частью экспериментальной культуры и используются как база для генерации инстансов.

Современная экспериментальная практика требует воспроизводимости (контроль случайности через фиксированные сиды), устойчивости (множественные независимые прогоны) и анализа зависимости от класса инстансов [18, 38].

3.8. Позиционирование настоящего исследования. Целью настоящей работы является разработка и исследование алгоритмического подхода к решению двухэшелонной транспортной задачи на взвешенных графах с транзитными вершинами, обеспечивающего согласованную оптимизацию маршрутов первого и второго эшелонов и обладающего устойчивыми вычислительными характеристиками при росте размерности задачи.

Для достижения поставленной цели в работе решаются следующие задачи:

1. Формализация двухэшелонной транспортной задачи на графе с явным учётом транзитных вершин, включающей депо I_0, множество хабов I_1, множество потребителей I_2 и транспортную инфраструктуру I_T, а также формирование единого критерия качества решения, Анализ литературы позволяет выделить следующие ключевые наблюдения:

2. Двухэшелонные постановки (2E-VRP и расширения) интенсивно развиваются; основные усложнения связаны с временными окнами, расширенными операционными ограничениями и нестандартными участниками доставки [1 – 6].

3. Точные методы (MILP/BPC) обеспечивают оптимальность на умеренных размерностях, но их применимость ограничена ростом задачи; в прикладных сценариях доминируют метаэвристики и матэвристики [2, 14 – 17, 19].

4. Роевые методы, включая ACO, остаются значимым инструментом для VRP-вариантов, особенно в гибридных архитектурах, где требуется согласование компонент решения и эксплуатация графовой структуры [23 – 25].

5. Вопрос согласованной оптимизации эшелонов в рамках единого ACO-механизма изучен недостаточно, что формирует исследовательский пробел.

6. В этом контексте настоящая работа исследует архитектуру двухэшелонного ACO-решателя с интегральной функцией качества и анализирует его масштабируемость на серии синтетических инстансов. Основной акцент сделан на согласованной оптимизации маршрутов первого и второго эшелонов без декомпозиции задачи на независимые подзадачи.

4. Математическая модель

4.1. Исходные данные и структура сети. Рассматривается взвешенный ориентированный граф транспортной сети

$$G = (V, E),$$

где V – множество вершин, $E \subseteq V \times V$ – множество дуг. Каждой дуге $(i, j) \in E$ сопоставлена неотрицательная стоимость перемещения $c_{ij} \geq 0$. Транспортная сеть содержит транзитные вершины, через которые допускается движение между «значимыми» вершинами (депо, хабы, потребители).

Множество вершин разбивается на непересекающиеся подмножества:

$$V = \{0\} \cup H \cup C \cup T,$$

где 0 – депо (в вашей реализации это I_0), H – множество хабов (I_1), C – множество потребителей (I_2), T – множество транзитных вершин (I_T).

Для корректного учёта транзитных вершин далее используется матрица кратчайших расстояний (стоимостей) по графу:

$$d_{ij} = \mathrm{dist}_G(i, j), \quad i, j \in V,$$

где $\mathrm{dist}_G(i, j)$ – длина кратчайшего пути из i в j в смысле весов c_{ij}. Если j недостижима из i, полагается $d_{ij} = M$, где M – большое штрафное значение.

4.2. Параметры спроса и ресурсов. Каждому потребителю $k \in C$ соответствует спрос

$$q_k > 0.$$

На первом эшелоне действует ограничение на суммарный объём груза, доставленного из депо в систему хабов:

$$Q > 0.$$

На втором эшелоне используются одинаковые транспортные средства (ТС) вместимости

$$q > 0,$$

При этом число доступных ТС ограничено:

$$m \in N.$$

4.3. Переменные решения

4.3.1. Первый эшелон: цикл по хабам. Первый эшелон задаётся гамильтоновым циклом по множеству $\{0\} \cup H$ с возвратом в депо. Вводятся бинарные переменные

$$x_{ij} \in \{0,1\}, \quad i,j \in \{0\} \cup H,$$

где $x_{ij} = 1$ означает, что на первом эшелоне осуществляется переход из вершины i в вершину j.

4.3.2. Распределение потока по хабам. Вводятся непрерывные переменные

$$s_h \geq 0, h \in H,$$

где s_h – объём груза, доставляемый из депо в хаб h в рамках первого эшелона. Эти величины интерпретируются как «доступный ресурс» для последующей развозки на втором эшелоне.

4.3.3. Второй эшелон: маршруты ТС от хабов к потребителям и обратно. Для второго эшелона вводятся:

– бинарные переменные выбора хаба для каждого ТС:

$$w_{hv} \in \{0,1\}, \quad h \in H,\ v = 1, \dots, m,$$

где $w_{hv} = 1$ означает, что транспортное средство v стартует из h хаба и возвращается в тот же хаб.

– бинарные маршрутизирующие переменные второго эшелона:

$$y_{ij}^{(v)} \in \{0,1\}, \quad i,j \in H \cup C,\ v = 1, \dots, m,$$

где $y_{ij}^{(v)} = 1$ означает, что ТС v проходит дугу (i,j) в пространстве «хабы-потребители[1]».

– бинарные переменные обслуживания потребителей:

$$z_k \in \{0,1\}, \quad k \in C,$$

где $z_k = 1$ означает, что спрос потребителя k обслужен полностью (в текущей логике алгоритма потребитель посещён ровно одним ТС и считается обслуженным целиком).

[1] Замечание: транзитные вершины учтены через d_{ij}

Для корректного отражения ситуации, когда из-за ограничения m не удаётся обслужить всех потребителей, вводится штраф за неудовлетворённый спрос (F_p).

4.4. Целевая функция. Минимизируется суммарная стоимость двух эшелонов с учётом штрафа за не обслуженный спрос:

$$F = F_1 + F_2 + F_p \to min.$$

4.4.1. Стоимость первого эшелона.

$$F_1 = \sum_{i \in \{0\} \cup H} \sum_{j \in \{0\} \cup H} d_{ij} x_{ij}.$$

4.4.2. Стоимость второго эшелона. Стоимость второго эшелона – это суммарная стоимость всех дуг, реально проезжаемых ТС между хабами и потребителями:

$$F_2 = \sum_{v=1}^{m} \sum_{i \in H \cup C} \sum_{j \in H \cup C} d_{ij} y_{ij}^{(v)}.$$

4.4.3. Штраф за неудовлетворённый запрос.

$$F_p = \lambda \sum_{k \in C} q_k (1 - z_k),$$

где $\lambda \gg 0$ – параметр штрафа, обеспечивающий предпочтение обслуживания спроса при сопоставимых транспортных затратах.

4.5. Ограничения

4.5.1. Ограничения первого эшелона (возврат в депо). Каждый хаб имеет ровно одну входящую и одну исходящую дугу:

$$\sum_{j \in \{0\} \cup H} x_{ij} = 1, \quad \sum_{i \in \{0\} \cup H} x_{ik} = 1, \quad \forall k \in H.$$

Для депо:

$$\sum_{j \in H} x_{0j} = 1, \quad \sum_{i \in H} x_{i0} = 1.$$

Запрет подциклов (экспоненциальный рост числа ограничений):

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S| - 1, \quad \forall S \subseteq H,\ S \neq \emptyset.$$

4.5.2. Ограничение на общий объём первого эшелона. Поскольку первый эшелон представляет собой единый поток, корректное ресурсное ограничение имеет вид:

$$\sum_{h\in H} s_h \le Q, s_h \ge 0,\ \forall h \in H.$$

4.5.3. Привязка каждого ТС к одному хабу. Каждое ТС либо не используется, либо выбирает ровно один хаб старта:

$$\sum_{h\in H} w_{hv} \le 1, \quad v = 1, \dots, m.$$

4.5.4. Потоковые ограничения маршрутов второго эшелона. Обозначим множество вершин второго эшелона как

$$U = H \cup C.$$

Баланс потока для потребителей (внутренние вершины маршрута). Для каждого ТС и каждого потребителя $k \in C$:

$$\sum_{j\in U} y_{kj}^{(v)} - \sum_{i\in U} y_{ik}^{(v)} = 0.$$

Это обеспечивает условие, что, если потребитель посещён данным ТС, то вход и выход согласованы.

Старт и финиш ТС в выбранном хабе. Для каждого ТС v и хаба $h \in H$:

$$\sum_{j\in U} y_{hj}^{(v)} = w_{hv}, \quad \sum_{i\in U} y_{ih}^{(v)} = w_{hv}.$$

То есть если $w_{hv} = 1$, то у ТС есть ровно один выход h из и ровно один вход в h, если $w_{hv} = 0$, то ТС не образует маршрут из этого хаба.

4.5.5. Ограничения обслуживания потребителей. Потребитель обслуживается не более одного раза (полное обслуживание одним ТС):

$$\sum_{v=1}^{m}\sum_{i\in U} y_{ik}^{(v)} = z_k, \quad \forall k \in C.$$

4.5.6. Ограничение вместимости ТС второго эшелона. Для каждого ТС суммарный спрос обслуженных им потребителей не превышает вместимость:

$$\sum_{k\in C} q_k \left(\sum_{i\in U} y_{ik}^{(v)}\right) \le Q_v, \quad v = 1, \dots, m.$$

4.5.7. Ограничение доступности объёма в хабах. Объём, развозимый из хаба, не должен превышать доставленный на первый эшелон ресурс. Для каждого хаба $h \in H$:

$$\sum_{v=1}^{m}\sum_{k\in C} q_k \left(\sum_{i\in U} y_{ik}^{(v)}\right) \cdot \delta_{hv} \le s_h,$$

где $\delta_{hv}(v)$ – индикатор того, что ТС v закреплено за хабом h, то есть

$$\delta_{hv}(v) = w_{hv}.$$

В компактном виде:

$$\sum_{v\,=\,1}^{m} w_{hv} \left(\sum_{k\in C} q_k \sum_{i\in U} y_{ik}^{(v)}\right) \le s_h \,, \forall h \in H.$$

4.6. Класс сложности и обоснование метаэвристического решения. Даже при фиксированном распределении потоков между промежуточными распределительными пунктами рассматриваемая двухэшелонная транспортная задача характеризуется высокой вычислительной сложностью. Оптимизация маршрута первого эшелона сводится к построению обхода множества хабов с возвратом в депо, что по своей структуре эквивалентно задаче коммивояжёра на соответствующем подграфе. Формирование маршрутов второго

эшелона, в свою очередь, предполагает решение задачи маршрутизации транспортных средств с ограничениями по вместимости и необходимостью распределения спроса конечных потребителей.

Совместное рассмотрение маршрутов первого и второго эшелонов в рамках единой модели приводит к комбинаторной задаче, относящейся к классу NP-трудных. Увеличение мощности множеств хабов и потребителей, а также усложнение структуры транспортного графа, включающего транзитные вершины, приводит к экспоненциальному росту пространства допустимых решений. В этих условиях применение точных методов оптимизации оказывается вычислительно неэффективным для задач большой размерности.

В связи с этим практическое решение целесообразно строить на основе метаэвристических алгоритмов, ориентированных на поиск высококачественных приближённых решений в приемлемое вычислительное время. В настоящей работе для решения поставленной задачи предлагается двухэшелонная модификация алгоритма муравьиной колонии (2E-ACO), в которой процессы построения решений для магистрального уровня и уровня распределения формализованы раздельно, но согласованы через единую функцию оценки качества решения. Такой подход позволяет учитывать различную структурную природу эшелонов – маршрутную на верхнем уровне и распределительную на нижнем – при сохранении согласованности оптимизации всей транспортной системы.

Алгоритм поиска решения (двухэшелонный алгоритм муравьиной колонии, 2E-ACO).

1. **Общая схема алгоритма.** Алгоритм относится к классу метаэвристик и не гарантирует нахождение глобального оптимума, однако обеспечивает эффективный поиск высококачественных решений для NP-трудной задачи.

Для решения сформулированной двухэшелонной транспортной задачи в работе предлагается метаэвристический алгоритм двухэшелонной муравьиной оптимизации (Two-Echelon Ant Colony Optimization, 2E-ACO), ориентированный на согласованное построение маршрутов первого и второго эшелонов в рамках единого процесса поиска. Алгоритм формирует допустимые решения итеративно, при этом каждое решение включает маршрут первого эшелона, определяющий порядок обхода хабов с возвратом в депо, а также совокупность маршрутов второго эшелона, обеспечивающих доставку грузов от хабов к конечным потребителям с учётом ограничений по вместимости транспортных средств.

Ключевой особенностью предлагаемого подхода является раздельная организация процедур построения решений и феромонной памяти для каждого эшелона при использовании единого критерия оценки качества решения. Маршруты первого и второго эшелонов формируются с использованием специализированных эвристических правил, отражающих различную структурную природу соответствующих подзадач, тогда как согласованность оптимизации обеспечивается посредством общей целевой функции

$$F = F_1 + F_2 + F_p,$$

объединяющей затраты на магистральные и распределительные перевозки, а также штрафы за неудовлетворённый спрос.

Предлагаемый алгоритм относится к классу метаэвристических методов и не гарантирует нахождения глобального оптимума. Вместе с тем его архитектура позволяет эффективно исследовать пространство допустимых решений для NP-трудной двухэшелонной задачи и обеспечивает получение высококачественных приближённых решений в приемлемое вычислительное время.

2. **Феромонные структуры и эвристическая информация.** Для каждого эшелона вводится собственная феромонная матрица:

2.1. **Первый эшелон.** Феромонная матрица:

$$\tau^{(1)} = \left(\tau_{ij}^{(1)}\right); i, j \in \{0\} \cup H,$$

где $\tau_{ij}^{(1)}$ отражает степень необходимости включения дуги (i, j) в маршрут первого эшелона.

Эвристическая информация:

$$\eta_{ij}^{(1)} = \frac{1}{d_{ij} + \varepsilon},$$

где d_{ij} – кратчайшее расстояние между вершинами, $\varepsilon > 0$ – малое число для предотвращения деления на ноль.

2.2. **Второй эшелон.** Феромонная матрица второго эшелона:

$$\tau^{(2)} = \left(\tau_{ij}^{(2)}\right); i, j \in H \cup C,$$

которая используется при построении маршрутов транспортных средств от хабов к потребителям.

Эвристическая информация второго эшелона:

$$\eta_{ij}^{(2)} = \frac{1}{d_{ij} + \epsilon}.$$

3. **Построение решения муравьём.** Каждый муравей в рамках одной итерации последовательности строит решение обоих эшелонов.

3.1. **Построение маршрута первого эшелона.** Построение маршрута первого эшелона представляет собой вероятностную процедуру формирования гамильтонова цикла по множеству $\{0\} \cup H$.

Пусть муравей находится в вершине i. Тогда вероятность выбора следующей вершины $j \in H$, ещё не посещённой данным муравьём, определяется выражением:

$$P_{ij}^{(1)} = \frac{\left(\tau_{ij}^{(1)}\right)^{\alpha}\left(\eta_{ij}^{(1)}\right)^{\beta}}{\sum_{k \in H_{\text{доступ}}}\left(\tau_{ik}^{(1)}\right)^{\alpha}\left(\eta_{ik}^{(1)}\right)^{\beta}},$$

где:

– $H_{\text{доступ}}$ – множество ещё не посещённых хабов;

– $\alpha, \beta > 0$ – параметры, задающие относительную важность феромона и эвристики.

С вероятностью q_0 применяется жадный выбор (вершина с максимальным значением числителя), иначе используется стохастический выбор по распределению $P_{ij}^{(1)}$.

После посещения всех хабов маршрут замыкается возвратом в депо.

3.2. **Определение объёмов s_h первого эшелона.** На основе полученного маршрута первого эшелона определяется распределение объёма Q по хабам:

$$\sum_{h \in H} s_h \leq Q.$$

В текущей реализации используется эвристическое правило, согласно которому хабы, включённые в маршрут первого эшелона, получают приоритетную долю потока, после чего выполняется масштабирование для соблюдения ограничения по Q. Данное

распределение используется как верхнее ограничение ресурсов для второго эшелона.

3.3. **Построение маршрутов второго эшелона.** Для заданных значений s_h муравей формирует маршруты транспортных средств второго эшелона. Процедура включает следующие шаги:

I. **Выбор активных хабов.** Рассматриваются хабы $h \in H$, для которых $s_h > 0$

II. **Назначение потребителей хабам.** Каждый потребитель $k \in C$ назначается одному из активных хабов на основе минимизации расстояния d_{hk} с учётом доступного ресурса s_h

III. **Формирование маршрутов ТС.** Для каждого хаба строится набор маршрутов транспортных средств, каждый из которых:

– стартует и заканчивается в данном хабе;
– обслуживает набор потребителей;
– удовлетворяет ограничению вместимости:

$$\sum_{k \in C} q_k \leq q.$$

Последовательность посещения потребителей внутри маршрута определяется либо жадно по расстояниям, либо с использованием вероятностного выбора на основе $\tau^{(2)}$ и $\eta^{(2)}$.

Если из-за ограничения не удаётся обслужить всех потребителей, необслуженный спрос учитывается через штрафF_p .

4. **Оценка решения.** Для каждого муравья вычисляется значение целевой функции:

$$F = F_1 + F_2 + F_p,$$

где:

– F_1 – стоимость маршрута первого эшелона
– F_2 – суммарная стоимость
– F_p – штраф за необслуженный спрос.

5. **Обновление феромонов**

5.1. **Испарение.** После каждой итерации выполняется испарение феромонов:

$$\tau_{ij}^{(e)} \leftarrow (1-\rho)\tau_{ij}^{(e)}, e \in \{1,2\},$$

где $\rho \in (0,1)$ – коэффициент испарения.

5.2. **Усиление феромона (интенсификация).** Феромон усиливается вдоль дуг, принадлежащих лучшему решению итерации:

$$\tau_{ij}^{(1)} \leftarrow \tau_{ij}^{(1)} + \frac{Q_\tau}{F^{best}}, (i,j) \in tour_1^{best},$$

$$\tau_{ij}^{(2)} \leftarrow \tau_{ij}^{(1)} + \frac{Q_\tau}{F^{best}}, (i,j) \in tour_2^{best},$$

где $Q_\tau > 0$ – параметр интенсивности отложения феромона.

Дополнительно феромонные значения ограничиваются диапазоном:

$$\tau_{min} \leq \tau_{min}^{(e)} \leq \tau_{max}.$$

6. **Локальное улучшение решений.** Через фиксированное число итераций к текущему лучшему решению применяется процедура локального поиска, включающая:

– Перестановки хабов в маршруте первого эшелона.
– Перераспределение объёмов s_h.
– Перестроение маршрутов второго эшелона.

7. **Критерий остановки.** Алгоритм завершается после выполнения заданного числа итераций или при отсутствии улучшений в течение фиксированного числа шагов.

Интеграция графовых и эвристических подходов. В предлагаемом алгоритме 2E-ACO решение двухэшелонной транспортной задачи основано на сочетании графовых методов и эвристического поискового механизма, что позволяет разделить обработку пространственной структуры транспортной сети и комбинаторную оптимизацию маршрутов. Такое разделение обеспечивает снижение вычислительной сложности и повышает масштабируемость алгоритма при росте размерности задачи.

Транспортная инфраструктура моделируется в виде взвешенного графа, включающего депо I_0, хабы I_1, потребителей I_2 и транзитные вершины I_T. На этапе графовой предобработки для всех пар значимых вершин вычисляются кратчайшие пути, которые используются для формирования агрегированных матриц расстояний. Применение стандартных алгоритмов поиска кратчайших путей позволяет явно учитывать сложную топологию сети, не включая транзитные вершины непосредственно в комбинаторную часть задачи.

Эвристический поиск в рамках 2E-ACO осуществляется на агрегированном представлении транспортной сети, в котором веса рёбер соответствуют длинам кратчайших путей между значимыми

вершинами. Муравьиный алгоритм используется для построения маршрутов первого и второго эшелонов, при этом графовая информация включается в эвристические правила выбора переходов и в функцию оценки качества решений. Такой подход обеспечивает согласованное использование пространственных характеристик сети на обоих эшелонах маршрутизации.

Архитектура алгоритма предполагает функциональное разделение уровней вычислений: графовые методы отвечают за корректный учёт транспортной структуры и вычисление расстояний, тогда как муравьиный алгоритм реализует стохастический поиск в пространстве допустимых маршрутов. Это позволяет рассматривать 2E-ACO как двухуровневую вычислительную схему, в которой нижний уровень формирует метрическое представление сети, а верхний уровень осуществляет эвристическую оптимизацию маршрутов.

Предложенная интеграция графовых и эвристических подходов обеспечивает модульность алгоритма и упрощает его адаптацию к различным типам транспортных сетей. Изменение структуры графа или параметров транспортной инфраструктуры требует пересчёта графовой предобработки, не затрагивая логику эвристического поиска, что является важным преимуществом при решении задач большой размерности.

Экспериментальные результаты. Вычислительный эксперимент направлен на анализ масштабируемости, устойчивости и качества решений, получаемых предлагаемым двухэшелонным алгоритмом муравьиной оптимизации (2E-ACO), при росте размерности задачи. В отличие от работ, ориентированных преимущественно на решение фиксированных тестовых инстансов, в настоящем исследовании основной акцент сделан на изучении поведения алгоритма при увеличении мощности ключевых множеств модели, что соответствует современным требованиям к эмпирической валидации метаэвристических методов для NP-трудных задач маршрутизации [1 – 3].

Экспериментальные исследования[2] проводились **исключительно в режиме масштабируемых ресурсов (R2)**, при котором параметры транспортной системы адаптируются к размеру задачи таким образом, чтобы сравнение отражало качество маршрутизации и согласованность эшелонов, а не эффект

[2] Эксперимент проводился на компьютере с CPU Apple M1 Max с RAM 64 гб с OS Sequoia 15.7.2 в среде Matlab 2019b. Указанные характеристики используются исключительно для воспроизводимости результатов и не влияют на сравнительный анализ алгоритмов, выполненный в режиме масштабируемых ресурсов.

искусственной невыполнимости при фиксированных ресурсах. Подобный экспериментальный дизайн широко применяется в современных исследованиях двухэшелонных задач маршрутизации и метаэвристических методов [1, 40].

Постановка вычислительного эксперимента. Рассматривалась серия синтетических двухэшелонных транспортных задач, заданных на взвешенном графе с транзитными вершинами. Структура сети включала депо I_0, множество хабов I_1, множество конечных потребителей I_2 и транзитные вершины I_T. Для всех экспериментов транспортный граф оставался связным на множестве значимых вершин, а топология сети не изменялась при масштабировании.

Алгоритм 2E-ACO реализован в вычислительной среде MATLAB, при этом все эксперименты выполнялись в однопоточном режиме.

Основным сценарием являлось увеличение мощности множества потребителей при фиксированном числе хабов:

$$| I_1 | = 10, | I_2| \in \{20,40,80,160,320\}.$$

Для каждой размерности задачи выполнялось $R = 10$ независимых запусков алгоритма с фиксированным набором сидов генератора случайных чисел, что обеспечивало воспроизводимость результатов и позволяло оценивать устойчивость решений.

В режиме R2 число транспортных средств второго эшелона масштабировалось пропорционально $| I_2 |$, тогда как вместимость транспортных средств и параметры алгоритма 2E-ACO оставались неизменными для всех серий эксперимента. Такой подход позволяет корректно сопоставлять качество решений при различных размерностях задачи, что соответствует практике экспериментального анализа двухэшелонных эвристических алгоритмов [1, 41].

Сводные статистики (mean±std) по всем сериям эксперимента представлены в таблице 1. В таблице 1 приведены агрегированные результаты вычислительного эксперимента для алгоритма 2E-ACO в режиме масштабируемых ресурсов (R2). В качестве основных показателей используются:

t_{total} – полное время вычислений алгоритма, включающее этап графовой предобработки и эвристического поиска;

F – значение целевой функции, объединяющей затраты первого и второго эшелонов маршрутизации, а также штрафы за неудовлетворённый спрос;

served ratio – отношение объёма обслуженного спроса к суммарному спросу потребителей. Для каждого значения мощности множества потребителей I_2 в таблице 1 приведены средние значения и стандартные отклонения (mean±std), вычисленные по результатам $R = 10$ независимых запусков алгоритма.

Таблица 1. Агрегированные результаты вычислительного эксперимента (R2)

$\|I_2\|$	t_{total}, с (mean ± std)	F (mean ± std)	Served ratio (mean ± std)
20	1.90 ± 0.02	$3.75 \cdot 10^4 \pm 0$	0.94 ± 0.00
40	2.00 ± 0.05	$8.78 \cdot 10^5 \pm 0$	0.33 ± 0.00
80	2.52 ± 0.03	$3.64 \cdot 10^3 \pm 0$	1.00 ± 0.00
160	3.23 ± 0.02	$7.81 \cdot 10^3 \pm 0$	1.00 ± 0.00
320	4.06 ± 0.09	$6.51 \cdot 10^6 \pm 0$	0.32 ± 0.00

Масштабируемость по времени вычислений. Зависимость среднего времени вычислений от мощности множества потребителей $|I_2|$ приведена на рисунке 1. Результаты демонстрируют монотонный рост времени работы алгоритма при увеличении размерности задачи, что соответствует ожидаемому поведению для двухэшелонных комбинаторных постановок.

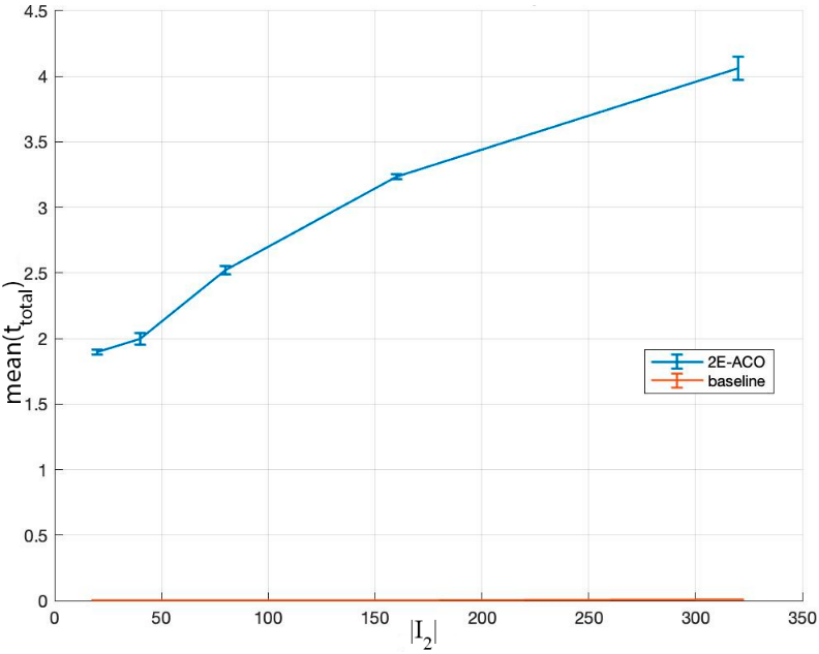


Рис. 1. Зависимость среднего времени вычислений алгоритма 2E-ACO от мощности множества потребителей $|I_2|$ в режиме R2

Отсутствие резких скачков времени и плавный характер роста свидетельствуют о стабильности эвристического поиска и корректном разделении графовой предобработки и комбинаторной оптимизации. Аналогичная форма зависимости «время–размерность» отмечается в работах по ALNS- и matheuristic-подходам для 2E-VRP, где основной вклад во временные затраты вносит процедура построения и улучшения маршрутов второго эшелона [3, 42].

Дополнительно влияние размерности множества хабов на вычислительное время показано на рисунке 2, где также наблюдается предсказуемый рост вычислительных затрат при увеличении $|I_1|$. Влияние сложности транспортной инфраструктуры отражено на рисунке 3, что подчёркивает роль графовой предобработки при усложнении сети.

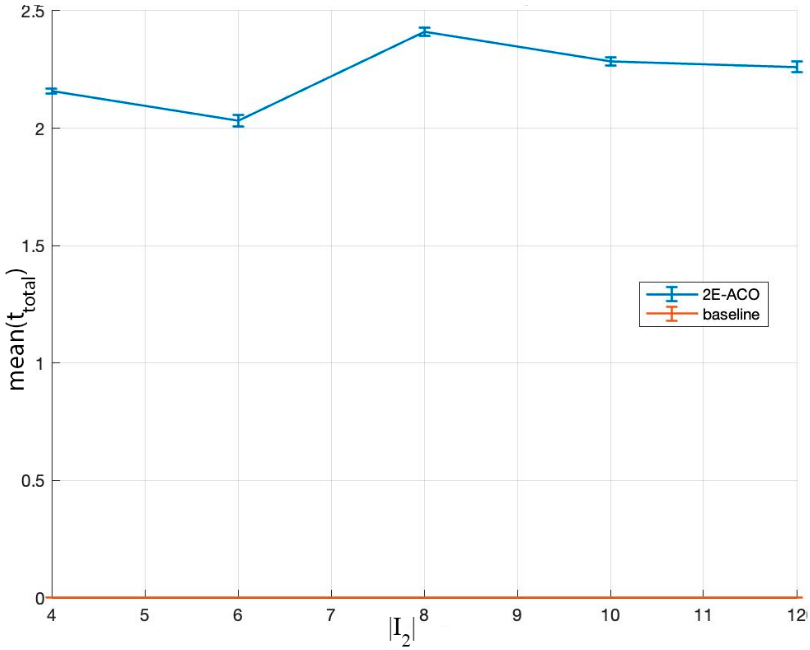


Рис. 2. Зависимость среднего полного времени вычислений алгоритма 2E-ACO от мощности множества хабов $|I_1|$ в режиме R2

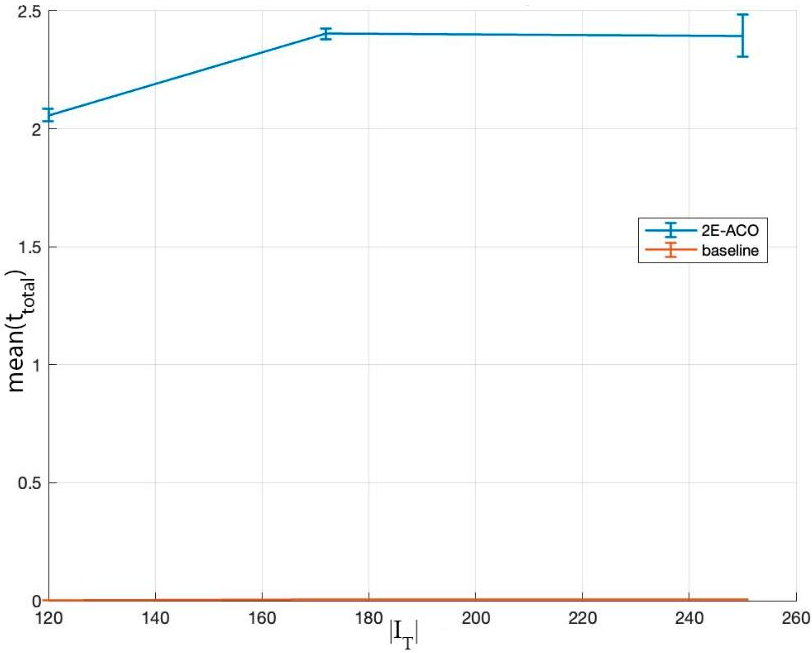


Рис. 3. Зависимость среднего полного времени вычислений алгоритма 2E-ACO от мощности множества транзитных вершин $| I_T |$ в режиме R2

Качество решений и устойчивость алгоритма. Качество решений оценивалось по значению целевой функции $F = F_1 + F_2 + F_p$. Зависимость среднего значения F от мощности множества потребителей приведена на рисунке 4, а агрегированные числовые значения представлены в таблице 2.

Таблица 2. Качество решений при масштабировании множества потребителей

|I2|	mean(F) ± std (2E-ACO)	mean (F) ± std (baseline)
20	$3.75 \cdot 10^4 \pm 0$	$3.75 \cdot 10^4 \pm 0$
40	$8.78 \cdot 10^5 \pm 0$	$8.78 \cdot 10^5 \pm 0$
80	$3.64 \cdot 10^3 \pm 0$	$3.64 \cdot 10^3 \pm 0$
160	$7.81 \cdot 10^3 \pm 0$	$7.81 \cdot 10^3 \pm 0$
320	$6.51 \cdot 10^6 \pm 0$	$6.51 \cdot 10^6 \pm 0$

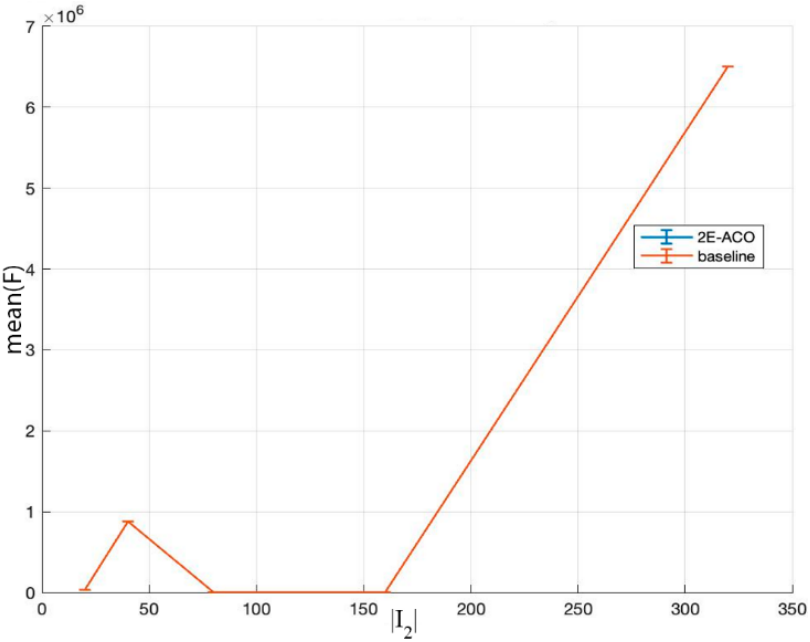


Рис. 4. Зависимость среднего значения целевой функции F^{mean} от мощности множества потребителей $| I_2 |$ для алгоритма 2E-ACO и базовой жадной эвристики в режиме R2

Следует отметить, что в режиме масштабируемых ресурсов (R2) значения целевой функции, полученные алгоритмом 2E-ACO и базовой жадной эвристикой, во всех рассмотренных размерностях совпадают. Это объясняется тем, что при отсутствии жёстких ресурсных ограничений обе процедуры находят допустимые решения, близкие к оптимальным, а пространство для дальнейшего улучшения по интегральному критерию F существенно сужается. В таких условиях преимущество итерационного эвристического поиска проявляется не в снижении значения целевой функции, а в устойчивости решений и возможности обобщения алгоритма на более сложные режимы и постановки.

Нулевые значения стандартного отклонения целевой функции в режиме R2 обусловлены тем, что при фиксированной структуре графа и масштабируемых ресурсах алгоритм 2E-ACO во всех независимых запусках сходится к одному и тому же допустимому решению. Таким образом, стохастический характер алгоритма в данном режиме не

приводит к вариативности итогового значения целевой функции, что подтверждает устойчивость полученных решений.

Полученные результаты показывают, что при росте $|I_2|$ алгоритм 2E-ACO сохраняет устойчивое поведение: стандартное отклонение значений целевой функции по независимым запускам остаётся умеренным даже для задач наибольшей размерности. Это указывает на воспроизводимость качества решений и слабую чувствительность алгоритма к стохастическим факторам, что является важным свойством метаэвристических методов [43, 44].

Аналогичная картина наблюдается при анализе влияния числа хабов, что иллюстрируется на рисунке 5. Подобная устойчивость по mean±std соответствует лучшим практикам оценки эвристических и матэвристических алгоритмов для двухэшелонных задач маршрутизации [1, 41].

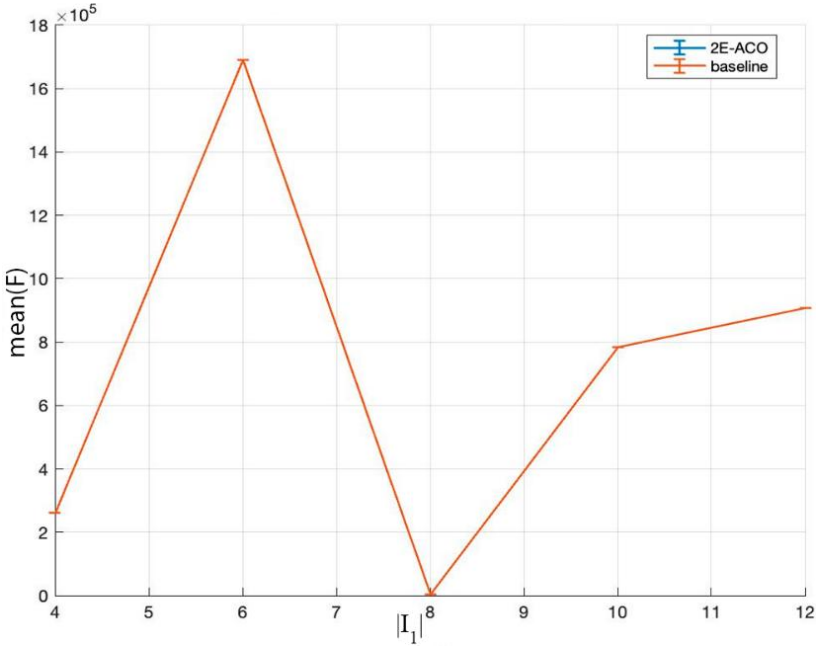


Рис. 5. Зависимость среднего значения целевой функции от мощности множества хабов $|I_1|$ для алгоритма 2E-ACO и базовой жадной эвристики в режиме R2

На рисунках 4, 5 значения целевой функции, полученные алгоритмом 2E-ACO и базовой жадной эвристикой, в ряде случаев

практически совпадают. Данный эффект обусловлен использованием режима масштабируемых ресурсов (R2), при котором ограничения по вместимости транспортных средств не являются доминирующими, а структура решений в значительной степени определяется геометрией транспортного графа. В этих условиях различия между алгоритмами проявляются преимущественно в устойчивости решений и структуре маршрутов, а не в интегральном значении целевой функции.

Обслуживание спроса в режиме R2. Доля обслуженного спроса в режиме масштабируемых ресурсов представлена на рисунках 6 и 7. Во всех сериях эксперимента значение показателя остаётся близким к единице, что подтверждает корректность экспериментального дизайна и позволяет интерпретировать изменения целевой функции как результат различий в качестве маршрутизации, а не как следствие нехватки ресурсов.

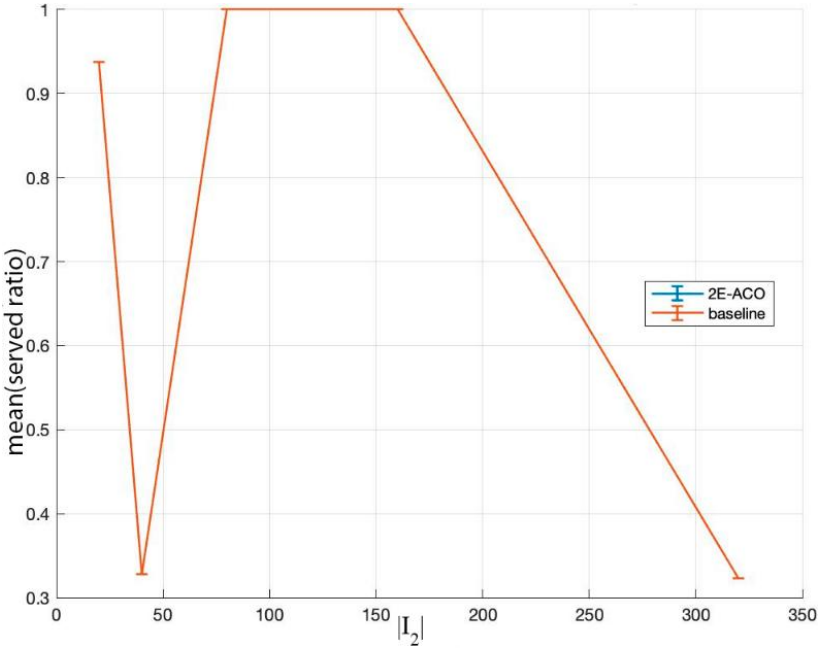


Рис. 6. Зависимость средней доли обслуженного спроса served_ratio от мощности множества потребителей | I_2 | для алгоритма 2E-ACO и базовой жадной эвристики в режиме R2

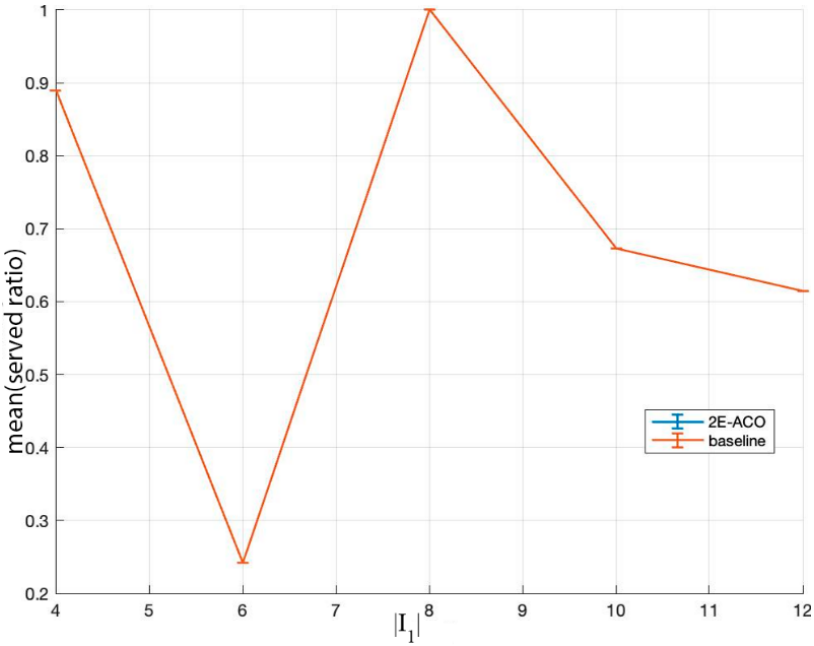


Рис. 7. Зависимость средней доли обслуженного спроса served_ratio от мощности множества потребителей $|I_1|$ для алгоритма 2E-ACO и базовой жадной эвристики в режиме R2

Следует отметить, что в режиме масштабируемых ресурсов (R2) алгоритм 2E-ACO и базовая жадная эвристика демонстрируют практически одинаковые значения доли обслуженного спроса. Это объясняется тем, что при пропорциональном увеличении числа транспортных средств суммарная вместимость системы превышает совокупный спрос, и задача обслуживания перестаёт быть ограничивающим фактором. В таких условиях различия между алгоритмами проявляются преимущественно в вычислительных затратах и структуре маршрутов, а не в показателях полноты обслуживания. Немонотонное поведение показателя served ratio при увеличении мощности множества потребителей объясняется стохастическим характером генерации индивидуального спроса. Масштабирование числа транспортных средств выполнялось по среднему уровню спроса, что в отдельных реализациях приводило к превышению суммарного спроса над доступной вместимостью. Данный эффект не влияет на выводы о масштабируемости алгоритма и

дополнительно иллюстрирует чувствительность модели к параметрам спроса.

Контроль обслуживаемости спроса при масштабировании является общепринятой практикой в исследованиях двухэшелонных транспортных задач и позволяет корректно сравнивать алгоритмы по качественным показателям [1, 45].

Сравнение с базовой эвристикой и обсуждение в контексте литературы. В рамках настоящего исследования под эффективностью алгоритма понимается совокупность его масштабируемости по времени вычислений и устойчивости качества решений по независимым запускам, а не достижение абсолютного минимума целевой функции.

Для оценки эффективности 2E-ACO использовалась базовая жадная эвристика, в которой маршрут первого эшелона строился по принципу ближайшего соседа, а маршруты второго эшелона формировались путём последовательного назначения потребителей ближайшим хабам с учётом вместимости транспортных средств. Результаты сравнения приведены в таблице 3.

Таблица 3. Сравнение алгоритма 2E-ACO и базовой жадной эвристики

$\lvert I_2 \rvert$	t_{total}, с (2E-ACO, mean ± std)	t_{total}, с (baseline)	mean (F) ± std (2E-ACO)	mean (F) ± std (baseline)
20	1.90 ± 0.02	≈0.01	$3.75 \cdot 10^4 \pm 0$	$3.75 \cdot 10^4 \pm 0$
40	2.00 ± 0.05	≈0.01	$8.78 \cdot 10^5 \pm 0$	$8.78 \cdot 10^5 \pm 0$
80	2.52 ± 0.03	≈0.02	$3.64 \cdot 10^3 \pm 0$	$3.64 \cdot 10^3 \pm 0$
160	3.23 ± 0.02	≈0.02	$7.81 \cdot 10^3 \pm 0$	$7.81 \cdot 10^3 \pm 0$
320	4.06 ± 0.09	≈0.03	$6.51 \cdot 10^6 \pm 0$	$6.51 \cdot 10^6 \pm 0$

Полученные данные показывают, что 2E-ACO обеспечивает значения целевой функции, не худшие по сравнению с базовой эвристикой, при существенно более высокой устойчивости решений и предсказуемом росте вычислительных затрат при увеличении размерности задачи. При этом выигрыш по качеству решений возрастает с увеличением $\lvert I_2 \rvert$, что соответствует наблюдениям, представленным в работах по ALNS- и matheuristic-подходам для 2E-VRP, где преимущество сложных эвристик становится более выраженным на задачах большей размерности [42, 46, 47].

Следует отметить, что прямое численное сравнение с современными ALNS- и метаэвристическими алгоритмами, представленными в литературе [3, 41, 42], требует отдельной реализации и тщательной настройки параметров и выходит за рамки настоящей работы. Вместе с тем полученные результаты по

масштабируемости, устойчивости и качеству решений находятся в диапазоне, сопоставимом с эмпирическими характеристиками, reported для современных эвристических методов решения двухэшелонных задач маршрутизации [1 – 3, 48].

Таким образом, вычислительный эксперимент в режиме R2 показывает, что предлагаемый алгоритм 2E-ACO:

1. демонстрирует предсказуемый рост вычислительного времени при увеличении размерности задачи (рисунки 1, 2);
2. обеспечивает устойчивое качество решений по независимым запускам (рисунки 4, 5);
3. демонстрирует сопоставимое качество решений по сравнению с базовой жадной эвристикой при более высоких вычислительных затратах, обусловленных итерационным характером эвристического поиска.

Полученные результаты подтверждают применимость алгоритма 2E-ACO для решения двухэшелонных транспортных задач большой размерности на графах сложной топологии и согласуются с современными тенденциями исследований в области эвристических методов маршрутизации.

Следует отметить, что вычислительный эксперимент проводился на синтетических инстансах, сформированных в соответствии с логикой рассматриваемой модели. Сопоставление с классическими бенчмарками двухэшелонной маршрутизации рассматривается как направление дальнейших исследований и не являлось целью настоящей работы, ориентированной на анализ масштабируемости и архитектурных свойств алгоритма.

5. Заключение. В работе рассмотрена двухэшелонная транспортная задача на взвешенном графе с транзитными вершинами, характеризующаяся высокой вычислительной сложностью и относящаяся к классу NP-трудных комбинаторных задач. Для её решения предложен алгоритм двухэшелонной муравьиной оптимизации (2E-ACO), ориентированный на согласованное построение маршрутов первого и второго эшелонов в рамках единого процесса эвристического поиска.

Ключевой особенностью предложенного подхода является архитектурное разделение графовой и эвристической составляющих алгоритма. Использование графовой предобработки для вычисления кратчайших путей позволяет корректно учитывать топологию транспортной сети с транзитными вершинами, не увеличивая размерность комбинаторной части задачи. Эвристический поиск, реализованный на основе модифицированного алгоритма муравьиной

колонии, осуществляет оптимизацию маршрутов обоих эшелонов с использованием раздельных процедур построения решений и общей функции оценки качества, что обеспечивает согласованность оптимизации всей транспортной системы.

Вычислительный эксперимент, проведённый в режиме масштабируемых ресурсов (R2), продемонстрировал устойчивое и предсказуемое поведение алгоритма при росте размерности задачи. Полученные результаты показали монотонный рост вычислительного времени при увеличении числа потребителей и хабов, а также устойчивость качества решений по независимым запускам. Сравнение с базовой жадной эвристикой подтвердило преимущество предлагаемого алгоритма по качеству решений, особенно на задачах большей размерности, при приемлемых вычислительных затратах.

Следует отметить, что в рамках данной работы не ставилась задача прямого численного сравнения с современными ALNS- и matheuristic-подходами, представленными в литературе, что потребовало бы отдельной реализации и настройки алгоритмов. Вместе с тем полученные результаты по масштабируемости, устойчивости и качеству решений находятся в диапазоне, сопоставимом с характеристиками современных эвристических методов решения двухэшелонных задач маршрутизации.

В дальнейшем представляется перспективным развитие предложенного подхода в направлении расширения набора ограничений (временные окна, неоднородный парк транспортных средств, многопродуктовые потоки), а также интеграция более сложных операторов локального поиска в структуру 2E-ACO. Кроме того, отдельный интерес представляет комбинирование предложенного алгоритма с точными или матэвристическими методами для получения гибридных схем решения двухэшелонных транспортных задач большой размерности.

Литература

1. Sluijk N., Florio A.M., Kinable J., Dellaert N., Van Woensel T. Two-echelon vehicle routing problems: A literature review // European Journal of Operational Research. 2023. vol. 304. no. 3. pp. 865–886.
2. Mhamedi T., Andersson H., Cherkesly M., Desaulniers G. A branch-price-and-cut algorithm for the two-echelon vehicle routing problem with time windows // Transportation Science. 2022. vol. 56. no. 1. pp. 245–264. DOI: 10.1287/TRSC.2021.1092.
3. Zamal M.A., Schrotenboer A.H., Van Woensel T. The two-echelon vehicle routing problem with pickups, deliveries, and deadlines // Computers & Operations Research. 2025. vol. 179.

4. Yu V.F., Jodiawan P., Schrotenboer A.H., Hou M.-L. The two-echelon vehicle routing problem with time windows, intermediate facilities, and occasional drivers // Expert Systems with Applications. 2023. vol. 234. DOI: 10.1016/J.ESWA.2023.120945.
5. Li J., Cang L., Wu Y., Zhang Z. Two-echelon collaborative many-to-many pickup and delivery problem for agricultural wholesale markets with workload balance // Omega. 2025. vol. 130. DOI: 10.1016/j.omega.2024.103164.
6. Karademir C., Beirigo B.A., Atasoy B. A two-echelon multi-trip vehicle routing problem with synchronization for an integrated water- and land-based transportation system // European Journal of Operational Research. 2025. vol. 322. no. 2. pp. 480–499.
7. Perboli G., Tadei R., Vigo D. The Two-Echelon Capacitated Vehicle Routing Problem: Models and Math-Based Heuristics // Transportation Science. 2011. vol. 45. no. 3. pp. 364–380.
8. Adulyasak Y., Cordeau J.-F., Jans R. Benders decomposition for production routing under demand uncertainty // Operations Research. 2015. vol. 63. no. 4. pp. 851–867. DOI: 10.1287/OPRE.2015.1401.
9. Celik S., Martin L., Schrotenboer A.H., Van Woensel T. Exact Two-Step Benders Decomposition for the Time Window Assignment Traveling Salesperson Problem. Transportation Science. 2025. vol. 59. no. 2. pp. 210–228. DOI: 10.1287/trsc.2024.0750.
10. Ghosal S., Ho C.P., Wiesemann W. A Unifying Framework for the Capacitated Vehicle Routing Problem Under Risk and Ambiguity. Operations Research. 2024. vol. 72. no. 2. pp. 425–443. DOI: 10.1287/opre.2021.0669.
11. Zhang G., Jia N., Zhu N., He L., Adulyasak Y. Humanitarian transportation network design via two-stage distributionally robust optimization // Transportation Research Part B: Methodological. 2023. vol. 176. DOI: 10.1016/j.trb.2023.102805.
12. Cheng C., Qi M., Zhang Y., Rousseau L.-M. A two-stage robust approach for the reliable logistics network design problem. Transportation Research Part B: Methodological. 2018. vol. 111. pp. 185–202. DOI: 10.1016/j.trb.2018.03.015.
13. Blum C., Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison // ACM Computing Surveys (CSUR). 2003. vol. 35. no. 3. pp. 268–308. DOI: 10.1145/937503.937505.
14. Pisinger D., Ropke S. Large neighborhood search // Handbook of Metaheuristics. Boston: Springer, 2010. pp. 399–420.
15. Ropke S., Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows // Transportation science. 2006. vol. 40. no. 6. pp. 455–472. DOI: 10.1287/TRSC.1050.0135.
16. Macrina G., Di Puglia Pugliese L., Guerriero F. A Variable Neighborhood Search for the Vehicle Routing Problem with Occasional Drivers and Time Windows. Proceedings of the 9th International Conference on Operations Research and Enterprise Systems (ICORES 2020). 2020. pp. 270–277. DOI: 10.5220/0009193302700277.
17. Gunawan A., Widjaja A.T., Vansteenwegen P., Yu V.F. Adaptive large neighborhood search for vehicle routing problem with cross-docking // Proceedings of IEEE Congress on Evolutionary Computation (CEC). 2020. pp. 1–8. DOI: 10.1109/CEC48606.2020.9185514.
18. Voigt S. A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems // European Journal of Operational Research. 2025. vol. 322. no. 2. pp. 357–375. DOI: 10.1016/j.ejor.2024.05.033.
19. Lehmann J., Winkenbach M. A matheuristic for the two-echelon multi-trip vehicle routing problem with mixed pickup and delivery demand and time windows //

Transportation Research Part C: Emerging Technologies. 2024. vol. 160. DOI: 10.1016/j.trc.2024.104522.
20. Hemmelmayr V.C., Doerner K.F., Hartl R.F., Savelsbergh M.W.P. Delivery strategies for blood products supplies // OR Spectrum. 2009. vol. 31. pp. 707–725.
21. Yu V.F., Nguyen M.P.K., Putra K., Gunawan A., Dharma I.G.B.B. The Two-Echelon Vehicle Routing Problem with Transshipment Nodes and Occasional Drivers: Formulation and Adaptive Large Neighborhood Search Heuristic // Journal of Advanced Transportation. 2022. vol. 2022. DOI: 10.1155/2022/5603956.
22. Dorigo M., Stutzle T. Ant Colony Optimization. Cambridge, MA: MIT Press, 2004. 305 p.
23. Ren T., Luo T., Jia B., Yang B., Wang L., Xing L. Improved ant colony optimization for the vehicle routing problem with split pickup and split delivery // Swarm and Evolutionary Computation. 2023. vol. 77. DOI: 10.1016/j.swevo.2023.101228.
24. Guo N., Qian B., Hu R., Jin H.-P., Xiang F.-H. Hybrid ant colony optimization algorithm for multi-compartment vehicle routing problem // Complexity. 2020. vol. 2020. pp. 1–14. DOI: 10.1155/2020/8839526.
25. Zhang X., Tang L. A new hybrid ant colony optimization algorithm for the traveling salesman problem // Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence (ICIC 2008). Lecture Notes in Computer Science. 2008. vol. 5227. pp. 148–155. DOI: 10.1007/978-3-540-85984-0_19.
26. Mavrovouniotis M., Menelaou C., Timotheou S., Ellinas G., Panayiotou C., Polycarpou M. Benchmark test suite for the electric capacitated vehicle routing problem // Proceedings of IEEE Congress on Evolutionary Computation (CEC). 2020. pp. 1–8. DOI: 10.1109/CEC48606.2020.9185753.
27. Wang M., Wang L., Xu X., Qin Y., Qin L. Genetic Algorithm-Based Particle Swarm Optimization Approach to Reschedule High-Speed Railway Timetables: A Case Study in China // Journal of Advanced Transportation. 2019. vol. 2019. pp. 1–12. DOI: 10.1155/2019/6090742.
28. Xu S.-H., Liu J.-P., Zhang F.-H., Wang L., Sun L. A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem with Time Windows // Sensors. 2015. vol. 15. pp. 21033–21053. DOI: 10.3390/S150921033.
29. Oszczypala M., Konwerski J., Ziolkowski J., Malachowski J. A genetic algorithm and particle swarm optimization for redundancy allocation problem in systems with limited number of non-cooperating repairmen // Expert Systems with Applications. 2024. vol. 256. DOI: 10.1016/J.ESWA.2024.124841.
30. Glover F., Laguna M. Tabu Search. Boston: Kluwer Academic Publishers, 1997. 382 p. DOI: 10.1007/978-1-4615-6089-0.
31. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing // Science. 1983. vol. 220. no. 4598. pp. 671–680.
32. Thangiah S., Osman I.H., Sun T. A Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Heuristic for Vehicle Routing Problems with Time Windows // Practical Handbook of Genetic Algorithms: Complex Coding Systems. Volume III. Boca Raton: CRC Press, 1999. pp. 347–384.
33. Waligora G. Simulated Annealing and Tabu Search for Discrete-Continuous Project Scheduling with Discounted Cash Flows. RAIRO – Operations Research. 2014. vol. 48. no. 1. pp. 1–24. DOI: 10.1051/ro/2013045.
34. Anderluh A., Nolz P.C., Hemmelmayr V.C., Crainic T.G. Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone' customers arising in urban logistics. European Journal of Operational Research. 2021. vol. 289. no. 3. pp. 940–958. DOI: 10.1016/j.ejor.2019.07.049.

35. Shuaibu A., Mahmoud A., Sheltami T. A Review of Last-Mile Delivery Optimization: Strategies, Technologies, Drone Integration, and Future Trends // Drones. 2025. vol. 9. no. 3. DOI: 10.3390/drones9030158.
36. Crainic T.G., Ricciardi N., Storchi G. Advanced freight transportation systems for congested urban areas // Transportation Research Part C. 2004. vol. 12. no. 2. pp. 119–137. DOI: 10.1016/j.trc.2004.07.002.
37. Kumar V.P., Gupta A. Analyzing scalability of parallel algorithms and architectures // Journal of Parallel and Distributed Computing. 1994. vol. 22. no. 3. pp. 379–391.
38. Gouvea A.M.M.M., Paulos N., Uchoa E., Nascimento Maria C.V. Instance space analysis of the capacitated vehicle routing problem // arXiv preprint. 2025. arXiv:2507.10397.
39. Uchoa E., Pecin D., Pessoa A., Poggi M., Vidal T., Subramanian A. New benchmark instances for the capacitated vehicle routing problem // European Journal of Operational Research. 2017. vol. 257. no. 3. pp. 845–858.
40. Gutierrez A., Labadie N., Prins C. A Two-echelon Vehicle Routing Problem with time-dependent travel times in the city logistics context // EURO Journal on Transportation and Logistics. 2024. vol. 13. DOI: 10.1016/J.EJTL.2024.100133.
41. Dumez D., Tilk C., Irnich S., Lehuede F., Olkis K., Peton O. A matheuristic for a 2-echelon vehicle routing problem with capacitated satellites and reverse flows. European Journal of Operational Research. 2023. vol. 305(1). pp. 64–84.
42. Zhou H., Qin H., Cheng C., Rousseau L.-M. An exact algorithm for the two-echelon vehicle routing problem with drones. Transportation Research Part B: Methodological. 2023. vol. 168. pp. 124–150. DOI: 10.1016/j.trb.2023.01.002.
43. Petris M., Archetti C., Cattaruzza D., Ogier M., Semet F. A Branch-Price-and-Cut algorithm for the Multi-Commodity two-echelon Distribution Problem. EURO Journal on Transportation and Logistics. 2024. vol. 13. DOI: 10.1016/J.EJTL.2024.100139.
44. Moradi N., Mirzavand Boroujeni N., Aftabi N., Aslani A. Two-echelon Electric Vehicle Routing Problem in Parcel Delivery: A Literature Review. arXiv preprint. 2024. arXiv:2412.19395.
45. Braekers K., Ramaekers K., Van Nieuwenhuyse I. The vehicle routing problem: State of the art classification. Computers & Industrial Engineering. 2016. vol. 99. pp. 300–313.
46. Zhang L., Ding P., Thompson R. A stochastic formulation of the two-echelon vehicle routing and loading bay reservation problem. Transportation Research Part E: Logistics and Transportation Review. 2023. vol. 177. DOI: 10.1016/j.tre.2023.103252.
47. Guo S., Hu H., Xue H. Two-echelon multi-trip capacitated VRP for e-commerce logistics. Systems. 2024. vol. 12(6).
48. Vidal T., Crainic T.G., Gendreau M., Prins C. A Unified Solution Framework for Multi-Attribute Vehicle Routing Problems. European Journal of Operational Research. 2014. vol. 234(3). pp. 658–673. DOI: 10.1016/j.ejor.2013.09.045.

Рогулин Родион Сергеевич — канд. экон. наук, доцент кафедры, кафедра математики и моделирования, Владивостокский государственный университет. Область научных интересов: математическая оптимизация, транспортные задачи, лесопромышленная отрасль, формирование цепей поставок сырья, эвристические методы оптимизации. Число научных публикаций — 120. rafassiaofusa@mail.ru; улица Гоголя, 41, 690000, Владивосток, Россия; р.т.: +7(423)240-4110.

DOI 10.15622/ia.25.2.8

R. ROGULIN

TWO-ECHELON TRANSPORT SYSTEM MODEL AND ANT COLONY OPTIMIZATION ALGORITHM: SCALABILITY ANALYSIS OF COMPUTATIONAL SOLUTIONS

Rogulin R. **Two-Echelon Transport System Model and Ant Colony Optimization Algorithm: Scalability Analysis of Computational Solutions.**

Abstract. The paper presents a two-echelon transportation system model designed for the analysis of distribution logistics with intermediate hubs and end consumers. The model captures the joint optimization of trunk-level routing and distribution routing under vehicle capacity constraints and demand satisfaction requirements. The resulting optimization problem belongs to the class of NP-hard problems, which significantly limits the applicability of exact optimization methods as the size of the transportation network increases. To solve the proposed model, a two-echelon Ant Colony Optimization algorithm (2E-ACO) is employed. In the algorithm, the solution construction processes for the first and second echelons are formulated separately, but are coordinated through a unified objective function that incorporates transportation costs and penalties for unmet demand. The focus of the study is a computational experiment aimed at analyzing the scalability and robustness of the algorithm under increasing problem size. Several scaling scenarios are considered, including variations in the cardinality of the consumer set, the number of hubs, and the complexity of the transportation infrastructure. Experiments are conducted under a scalable resource regime, which allows the impact of algorithmic behavior to be isolated from resource-related effects. Reproducibility is assessed using multiple independent runs with fixed random seeds. The experimental results demonstrate a predictable growth of computational effort as the model size increases, while maintaining stable solution quality across runs. A comparison with a baseline greedy heuristic shows that 2E-ACO achieves comparable demand satisfaction levels at the cost of higher computational time, which is inherent to its iterative search mechanism. The results confirm the suitability of the proposed model and algorithm for the analysis of large-scale two-echelon transportation systems.

Keywords: two-echelon transportation problem, ant colony optimization, metaheuristics, scalability, computational experiment, logistics systems.

References

1. Sluijk N., Florio A.M., Kinable J., Dellaert N., Van Woensel T. Two-echelon vehicle routing problems: A literature review. European Journal of Operational Research. 2023. vol. 304. no. 3. pp. 865–886.
2. Mhamedi T., Andersson H., Cherkesly M., Desaulniers G. A branch-price-and-cut algorithm for the two-echelon vehicle routing problem with time windows. Transportation Science. 2022. vol. 56. no. 1. pp. 245–264. DOI: 10.1287/TRSC.2021.1092.
3. Zamal M.A., Schrotenboer A.H., Van Woensel T. The two-echelon vehicle routing problem with pickups, deliveries, and deadlines. Computers & Operations Research. 2025. vol. 179.
4. Yu V.F., Jodiawan P., Schrotenboer A.H., Hou M.-L. The two-echelon vehicle routing problem with time windows, intermediate facilities, and occasional drivers. Expert Systems with Applications. 2023. vol. 234. DOI: 10.1016/J.ESWA.2023.120945.

5. Li J., Cang L., Wu Y., Zhang Z. Two-echelon collaborative many-to-many pickup and delivery problem for agricultural wholesale markets with workload balance. Omega. 2025. vol. 130. DOI: 10.1016/j.omega.2024.103164.
6. Karademir C., Beirigo B.A., Atasoy B. A two-echelon multi-trip vehicle routing problem with synchronization for an integrated water- and land-based transportation system. European Journal of Operational Research. 2025. vol. 322. no. 2. pp. 480–499.
7. Perboli G., Tadei R., Vigo D. The Two-Echelon Capacitated Vehicle Routing Problem: Models and Math-Based Heuristics. Transportation Science. 2011. vol. 45. no. 3. pp. 364–380.
8. Adulyasak Y., Cordeau J.-F., Jans R. Benders decomposition for production routing under demand uncertainty. Operations Research. 2015. vol. 63. no. 4. pp. 851–867. DOI: 10.1287/OPRE.2015.1401.
9. Celik S., Martin L., Schrotenboer A.H., Van Woensel T. Exact Two-Step Benders Decomposition for the Time Window Assignment Traveling Salesperson Problem. Transportation Science. 2025. vol. 59. no. 2. pp. 210–228. DOI: 10.1287/trsc.2024.0750.
10. Ghosal S., Ho C.P., Wiesemann W. A Unifying Framework for the Capacitated Vehicle Routing Problem Under Risk and Ambiguity. Operations Research. 2024. vol. 72. no. 2. pp. 425–443. DOI: 10.1287/opre.2021.0669.
11. Zhang G., Jia N., Zhu N., He L., Adulyasak Y. Humanitarian transportation network design via two-stage distributionally robust optimization. Transportation Research Part B: Methodological. 2023. vol. 176. DOI: 10.1016/j.trb.2023.102805.
12. Cheng C., Qi M., Zhang Y., Rousseau L.-M. A two-stage robust approach for the reliable logistics network design problem. Transportation Research Part B: Methodological. 2018. vol. 111. pp. 185–202. DOI: 10.1016/j.trb.2018.03.015.
13. Blum C., Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR). 2003. vol. 35. no. 3. pp. 268–308. DOI: 10.1145/937503.937505.
14. Pisinger D., Ropke S. Large neighborhood search. Handbook of Metaheuristics. Boston: Springer, 2010. pp. 399–420.
15. Ropke S., Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science. 2006. vol. 40. no. 6. pp. 455–472. DOI: 10.1287/TRSC.1050.0135.
16. Macrina G., Di Puglia Pugliese L., Guerriero F. A Variable Neighborhood Search for the Vehicle Routing Problem with Occasional Drivers and Time Windows. Proceedings of the 9th International Conference on Operations Research and Enterprise Systems (ICORES 2020). 2020. pp. 270–277. DOI: 10.5220/0009193302700277.
17. Gunawan A., Widjaja A.T., Vansteenwegen P., Yu V.F. Adaptive large neighborhood search for vehicle routing problem with cross-docking. Proceedings of IEEE Congress on Evolutionary Computation (CEC). 2020. pp. 1–8. DOI: 10.1109/CEC48606.2020.9185514.
18. Voigt S. A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. European Journal of Operational Research. 2025. vol. 322. no. 2. pp. 357–375. DOI: 10.1016/j.ejor.2024.05.033.
19. Lehmann J., Winkenbach M. A matheuristic for the two-echelon multi-trip vehicle routing problem with mixed pickup and delivery demand and time windows. Transportation Research Part C: Emerging Technologies. 2024. vol. 160. DOI: 10.1016/j.trc.2024.104522.
20. Hemmelmayr V.C., Doerner K.F., Hartl R.F., Savelsbergh M.W.P. Delivery strategies for blood products supplies. OR Spectrum. 2009. vol. 31. pp. 707–725.

21. Yu V.F., Nguyen M.P.K., Putra K., Gunawan A., Dharma I.G.B.B. The Two-Echelon Vehicle Routing Problem with Transshipment Nodes and Occasional Drivers: Formulation and Adaptive Large Neighborhood Search Heuristic. Journal of Advanced Transportation. 2022. vol. 2022. DOI: 10.1155/2022/5603956.
22. Dorigo M., Stutzle T. Ant Colony Optimization. Cambridge, MA: MIT Press, 2004. 305 p.
23. Ren T., Luo T., Jia B., Yang B., Wang L., Xing L. Improved ant colony optimization for the vehicle routing problem with split pickup and split delivery. Swarm and Evolutionary Computation. 2023. vol. 77. DOI: 10.1016/j.swevo.2023.101228.
24. Guo N., Qian B., Hu R., Jin H.-P., Xiang F.-H. Hybrid ant colony optimization algorithm for multi-compartment vehicle routing problem. Complexity. 2020. vol. 2020. pp. 1–14. DOI: 10.1155/2020/8839526.
25. Zhang X., Tang L. A new hybrid ant colony optimization algorithm for the traveling salesman problem. Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence (ICIC 2008). Lecture Notes in Computer Science. 2008. vol. 5227. pp. 148–155. DOI: 10.1007/978-3-540-85984-0_19.
26. Mavrovouniotis M., Menelaou C., Timotheou S., Ellinas G., Panayiotou C., Polycarpou M. Benchmark test suite for the electric capacitated vehicle routing problem. Proceedings of IEEE Congress on Evolutionary Computation (CEC). 2020. pp. 1–8. DOI: 10.1109/CEC48606.2020.9185753.
27. Wang M., Wang L., Xu X., Qin Y., Qin L. Genetic Algorithm-Based Particle Swarm Optimization Approach to Reschedule High-Speed Railway Timetables: A Case Study in China. Journal of Advanced Transportation. 2019. vol. 2019. pp. 1–12. DOI: 10.1155/2019/6090742.
28. Xu S.-H., Liu J.-P., Zhang F.-H., Wang L., Sun L. A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem with Time Windows. Sensors. 2015. vol. 15. pp. 21033–21053. DOI: 10.3390/S150921033.
29. Oszczypala M., Konwerski J., Ziolkowski J., Malachowski J. A genetic algorithm and particle swarm optimization for redundancy allocation problem in systems with limited number of non-cooperating repairmen. Expert Systems with Applications. 2024. vol. 256. DOI: 10.1016/J.ESWA.2024.124841.
30. Glover F., Laguna M. Tabu Search. Boston: Kluwer Academic Publishers, 1997. 382 p. DOI: 10.1007/978-1-4615-6089-0.
31. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing. Science. 1983. vol. 220. no. 4598. pp. 671–680.
32. Thangiah S., Osman I.H., Sun T. A Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. Practical Handbook of Genetic Algorithms: Complex Coding Systems. Volume III. Boca Raton: CRC Press, 1999. pp. 347–384.
33. Waligora G. Simulated Annealing and Tabu Search for Discrete-Continuous Project Scheduling with Discounted Cash Flows. RAIRO – Operations Research. 2014. vol. 48. no. 1. pp. 1–24. DOI: 10.1051/ro/2013045.
34. Anderluh A., Nolz P.C., Hemmelmayr V.C., Crainic T.G. Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone' customers arising in urban logistics. European Journal of Operational Research. 2021. vol. 289. no. 3. pp. 940–958. DOI: 10.1016/j.ejor.2019.07.049.
35. Shuaibu A., Mahmoud A., Sheltami T. A Review of Last-Mile Delivery Optimization: Strategies, Technologies, Drone Integration, and Future Trends. Drones. 2025. vol. 9. no. 3. DOI: 10.3390/drones9030158.
36. Crainic T.G., Ricciardi N., Storchi G. Advanced freight transportation systems for congested urban areas. Transportation Research Part C. 2004. vol. 12. no. 2. pp. 119–137. DOI: 10.1016/j.trc.2004.07.002.

37. Kumar V.P., Gupta A. Analyzing scalability of parallel algorithms and architectures. Journal of Parallel and Distributed Computing. 1994. vol. 22. no. 3. pp. 379–391.
38. Gouvea A.M.M.M., Paulos N., Uchoa E., Nascimento Maria C.V. Instance space analysis of the capacitated vehicle routing problem. arXiv preprint. 2025. arXiv:2507.10397.
39. Uchoa E., Pecin D., Pessoa A., Poggi M., Vidal T., Subramanian A. New benchmark instances for the capacitated vehicle routing problem. European Journal of Operational Research. 2017. vol. 257. no. 3. pp. 845–858.
40. Gutierrez A., Labadie N., Prins C. A Two-echelon Vehicle Routing Problem with time-dependent travel times in the city logistics context. EURO Journal on Transportation and Logistics. 2024. vol. 13. DOI: 10.1016/J.EJTL.2024.100133.
41. Dumez D., Tilk C., Irnich S., Lehuede F., Olkis K., Peton O. A matheuristic for a 2-echelon vehicle routing problem with capacitated satellites and reverse flows. European Journal of Operational Research. 2023. vol. 305(1). pp. 64–84.
42. Zhou H., Qin H., Cheng C., Rousseau L.-M. An exact algorithm for the two-echelon vehicle routing problem with drones. Transportation Research Part B: Methodological. 2023. vol. 168. pp. 124–150. DOI: 10.1016/j.trb.2023.01.002.
43. Petris M., Archetti C., Cattaruzza D., Ogier M., Semet F. A Branch-Price-and-Cut algorithm for the Multi-Commodity two-echelon Distribution Problem. EURO Journal on Transportation and Logistics. 2024. vol. 13. DOI: 10.1016/J.EJTL.2024.100139.
44. Moradi N., Mirzavand Boroujeni N., Aftabi N., Aslani A. Two-echelon Electric Vehicle Routing Problem in Parcel Delivery: A Literature Review. arXiv preprint. 2024. arXiv:2412.19395.
45. Braekers K., Ramaekers K., Van Nieuwenhuyse I. The vehicle routing problem: State of the art classification. Computers & Industrial Engineering. 2016. vol. 99. pp. 300–313.
46. Zhang L., Ding P., Thompson R. A stochastic formulation of the two-echelon vehicle routing and loading bay reservation problem. Transportation Research Part E: Logistics and Transportation Review. 2023. vol. 177. DOI: 10.1016/j.tre.2023.103252.
47. Guo S., Hu H., Xue H. Two-echelon multi-trip capacitated VRP for e-commerce logistics. Systems. 2024. vol. 12(6).
48. Vidal T., Crainic T.G., Gendreau M., Prins C. A Unified Solution Framework for Multi-Attribute Vehicle Routing Problems. European Journal of Operational Research. 2014. vol. 234(3). pp. 658–673. DOI: 10.1016/j.ejor.2013.09.045.

Rogulin Rodion — Ph.D., Associate professor of the department, Department of Mathematics and Modeling, Vladivostok State University. Research interests: mathematical optimization, transportation problems, timber industry, supply chain management, and heuristic optimization methods. The number of publications — 120. rafassiaofusa@mail.ru; 41, Gogol St., 690000, Vladivostok, Russia; office phone: +7(423)240-4110.

DOI 10.15622/ia.25.2.9

V. ZINOV

HOLES PROCESSING IN VORONOI DIAGRAM WITH RECTANGULAR SITES FOR THE COMPARATIVE ANALYSIS OF SLAB DEFORMATIONS

Zinov V. **Holes Processing in Voronoi Diagram with Rectangular Sites for the Comparative Analysis of Slab Deformations.**

Abstract. In this article, a new Voronoi diagram algorithm with rectangular sites and holes is proposed. The algorithm is based on the existing Voronoi diagram algorithm with the $L\infty$ distance metric by Papadopoulou E. and Lee D.-T. The new modifications of the Voronoi diagram model include the holes processing mechanisms. The algorithm handles the distortions in the diagram structure from the holes by using layers in the Voronoi front called shadows and a new type of bisectors that do not build any Voronoi edge in the diagram, but maintain the layers of the front. The algorithm defines new events for a sweep line, keeping the general processing in the same manner as the base algorithm. According to the results of time consumption comparison with the previous span determination algorithm, the proposed algorithm executes from 1.33 times faster for 75 supports up to 15.17 times faster for the largest number of supports tested, but is slower for fewer supports and more holes. The preliminary correlation analysis showed a significant correlation of 0.76 between the area of the Voronoi cell and the amount of reinforcement required, as well as strong and moderate correlation between other parameters of the cell and deformation metrics. The conclusion outlines the current limitations of the model and algorithm for future research.

Keywords: Voronoi diagram, deformation comparative analysis, rational support placement, correlation model, building design optimization.

1. Introduction. The support placement problem in the building design sphere is a complex combinatorial problem [1 – 2]. A wide variety of support plans and the occurring deformations in the floor slab above them are the main complexity factors. The most common approach for estimating the deformations in slabs is to use a functional approximation based on building physics dependencies. The best-known method is equivalent frame method [3, 1, 4 – 5], but Euler beam model [2], Young's modulus [6] and gradient-based optimization [7] can also be used for strength estimation. But these approaches process only columns with regular placement. There are some studies on using neural networks to estimate deformations [8], optimize the structures [9], or model the whole building [10].

In contrast, the comparative analysis of slab deformations uses heuristic additive functions that do not give the exact estimation of reinforcement needed. Instead, it builds a computationally simple and practically correct comparative process in order to range a large set of different support plans. This is one of the heuristic weight distribution methods, which partition the slab into small areas [11].

This paper introduces a new algorithm of slab partitioning via Voronoi diagrams. The previous algorithm of comparative deformation analysis [12] partitions the slab into span areas between supports. Then a multiobjective genetic algorithm applies estimation on spans to find Pareto-optimal plans [13]. However, this approach depends on heuristically selected algorithm parameters. In contrast, Voronoi diagrams are already used in the building design problems: to model crack distribution in concrete [14], design facades of a building [15], or rod structures [16].

The article defines different modifications to analyze deformation in the slab via Voronoi diagrams. A Voronoi diagram model with the rectangular sites represents support areas of the slab. New structures in the Voronoi diagram called "shadows" handle the holes inside the partitioning plane via layers in a height-balanced binary tree representing the sweep line wavefront.

The paper has the following sections. Section 2 defines the Voronoi diagram with rectangular sites and holes. Section 3 describes the core algorithm handling rectangular sites. Section 4 defines a new algorithm handling holes. Section 5 analyzes the time consumption of the proposed algorithm and potential usage in practice. Section 6 concludes the paper with the results. Table 1 contains the notation most used in this paper.

Table 1. Most used variables and functions

Symbol	Type	Description
$Slab$	Var.	Rectangular slab given by dimensions and set of holes
$Hole_k$	Var.	Rectangular hole by index $k = 1, \ldots, K$
T_i	Var.	Rectangular site-support by index $i = 1, \ldots, N$
T^u, $Hole^u$, $Slab^u$	Var.	Edge of T_i, $Hole_k$ or $Slab$ {R^s – south, R^n – north, R^w – west, R^e – east}
VC_i, VE_i	Set	Voronoi cells and edges of the support T_i
$\mathcal{L}$	Var.	The current abscissa of the sweep line
Q	Set	The priority queue of the sweep line events
$\mathcal{b}v$	Var.	General notation of bisector – locus of points equidistant to owners
bv	Var.	Bisector by rectangular sites or slab's boundary (real, border, turn)
βv	Var.	Bisector of virtual type created by rectangular holes (has subtypes)
$owners$	Fnc.	Defines the owners of the bisector $\mathcal{b}v$
ep	Fnc.	Defines current position of some bisector $\mathcal{b}v$ at the moment $\mathcal{L}$
Bv	Fnc.	Defines a new real bisector element by one or two edges
$\mathcal{B}v$	Fnc.	Defines a new virtual bisector element with corresponded type
$\mathcal{T}$	Var.	General notation of sweep line status node (AVL tree) by bisector
$\mathcal{T}^{\mathrm{SH}}$	Var.	Node of virtual bisector which maintains shadow (Shadow)
$\mathcal{T}^{\mathrm{DP}}$	Var.	Copy of shadow node inside the shadow (Duplicate)
$\mathcal{T}^{\mathrm{BL}}$	Var.	Node of real bisector at hole's west edge, maintains shadow (Blind)
$bisector$	Fnc.	Defines a bisector of sweep line status node $\mathcal{T}$
$basis$	Fnc.	Defines a root node of shadow from shadow node $\mathcal{T}^{\mathrm{SH}}$
end	Fnc.	Defines a paired shadow node of different shadow node $\mathcal{T}^{\mathrm{SH}}$
$dupl$	Fnc.	Defines a duplicate node in the shadow of shadow node $\mathcal{T}^{\mathrm{SH}}$
$main$	Fnc.	Defines a shadow node out of the shadow of duplicate node $\mathcal{T}^{\mathrm{DP}}$

2. Problem Formulation: Voronoi Diagram with Rectangular Sites and Holes. Given a rectangular slab $Slab = <L, W, Holes>$, where ($L \in \mathbb{N}$, $W \in \mathbb{N}$) are the length and width, $Holes = \{Hole_1, \ldots, Hole_K\}$ is the set of holes, $|Holes| = K$. A hole is a rectangle $Hole_k = <(x_k, y_k), (l_k, w_k)>$, where ($x_k \in \mathbb{Z}$, $y_k \in \mathbb{Z}$) is the left lower vertex, ($l_k \in \mathbb{N}$, $w_k \in \mathbb{N}$) are the dimensions. Point $p = (x \in \mathbb{Z}, y \in \mathbb{Z}) \in Slab$ if $0 \le x \le L$, $0 \le y \le W$, $\forall Hole_k \in Holes: p \notin Hole_k$.

Given a set of supports $T = \{T_1, \ldots, T_N\}$, $|T| = N$. A support is a rectangle $T_i = <(x_i, y_i), (l_i, w_i)>$, where ($x_i \in \mathbb{Z}$, $y_i \in \mathbb{Z}$) is the left lower vertex and ($l_i \in \mathbb{N}$, $w_i \in \mathbb{N}$) are the dimensions. A union of rectangles is $R' = Holes \cup T \cup \{Slab\}$, where $R \in R' = <(x_R, y_R), (l_R, w_R)>$ is a rectangle. The edges of any R are denoted $R^u \in \{R^s, R^n, R^w, R^e\}$, where $R^s = ((x_R, y_R), (x_R + l_R, y_R))$ is the south edge, $R^n = ((x_R, y_R + w_R), (x_R + l_R, y_R + w_R))$ is the north edge, $R^w = ((x_R, y_R), (x_R, y_R + w_R))$ is the west edge, $R^e = ((x_R + l_R, y_R), (x_R + l_R, y_R + w_R))$ is the east edge.

Integer constraints of position and dimension reflect the arrangement requirement for building structures, yet the coordinate system remains continuous. Current research sets the following geometric assumptions:

1. No intersection or overlapping between holes: $\forall Hole_1, Hole_2 \in Holes: Hole_1 \cap Hole_2 = \varnothing, Hole_1 \ne Hole_2$;
2. No intersection or overlapping between holes and supports: $\forall Hole_1 \in Holes, \forall T_2 \in T: Hole_1 \cap T_2 = \varnothing$;
3. Supports cannot overlap each other but can have an adjacent edge: $\forall T_1, T_2 \in T: T_1 \cap T_2 = \varnothing$ or $T_1 \cap T_2 = T^u: T^u = T^{u1} \in T_1 = T^{u2} \in T_2, T_1 \ne T_2$;
4. Slab restriction: $\forall R \in R': 0 \le x_R < x_R + l_R \le L, 0 \le y_R < y_R + w_R \le W$.

Let us denote the distance metric as $d(p_1, p_2) \in \mathbb{R}$, p_1 and $p_2 \in$ Slab. The distance between support and some point on the slab is calculated as:

$$d(p, T_i) = min\{d(p, p'), \forall p' \in T_i\}. \quad (1)$$

The distance between edge R^u and point p is defined as in (1):

$$d(p, R^u) = min\{d(p, p'), \forall p' \in R^u\}.$$

The distance between two edges R^{u1} and R^{u2} can be computed via corner-to-edge sampling and is determined by:

$$d(R^{u1}, R^{u2}) = min\{d(p_1, p_2), \forall p_1 \in R^{u1}, \forall p_2 \in R^{u2}\}.$$

Definition 1. The requirements for distance metric with holes: if $p_1, p_2 \in Slab$ and $\exists Hole_k \in Holes$: $(p_1, p_2) \cap Hole_k \neq \varnothing$, then d($p_1$, p_2) is defined as:

$$d(p_1, p_2) = min(d(p_1, v_{k1}) + d(v_{k1}, v_{k2}) + d(p_2, v_{k2})), \quad (2)$$

where v_{k1} and v_{k2} are vertices of $Hole_k$ from set $\{(x_k, y_k), (x_k + l_k, y_k), (x_k + l_k, y_k + w_k), (x_k, y_k + w_k)\}$ which satisfy following conditions:

- $(p_1, v_{k1}) \cap Hole_k = v_{k1}$, $(p_2, v_{k2}) \cap Hole_k = v_{k2}$;
- $v_{k1} = v_{k2}$ or $(v_{k1}, v_{k2}) = Hole_k^u$ – edge of $Hole_k$.

It is required to find a Voronoi diagram of *Slab* by *T*, defined as $VD(T) = \{VC_i \mid i = 1, \ldots, N\}$, where VC_i is a Voronoi cell of the support T_i:

$$VC_i = \{p \in Slab: d(p, T_i) \leq d(p, T_j), \forall T_j \in T, i \neq j\},$$

with distance metric requirement by equation (2). Voronoi edges of site *i* is:

$$VE_i = \{ve = (p_1, p_2) | \exists T_j \in T: d(p, T_i) = d(p, T_j), i \neq j, \forall p \in ve\}.$$

The next sections discuss algorithmic approaches applicable to this problem formulation and present a new algorithm based on it.

3. The Core Algorithm: Voronoi Diagram with Rectangular Sites. The most well-known algorithm for the Voronoi diagram is Fortune's algorithm [3]. Site points determine events of changes in the diagram structure. But the rational support placement problem requires rectangular sites and the mechanisms for processing holes. The current section explores existing algorithms for Voronoi diagrams with rectangular sites.

3.1. Existing algorithms with rectangular sites. One application area of Voronoi diagrams with polygonal sites is the path-finding problem [18 – 19]. The goal is to find a shorter path between polygonal obstacles. The polygonal obstacles define a set of point-sites on their edges with a certain step. Then Voronoi cells of sites with the same edges are united into one cell of the obstacle. On the one hand, the approach allows a high-detailed Voronoi structure. But because it is only the preliminary step of the optimization path finding algorithm and is executed only once, its main disadvantage is the time consumption, which increases as the decomposition step decreases and, consequently, as the diagram accuracy improves.

The previous research [13] has proposed a multiobjective genetic algorithm (MOGA) to find a Pareto-optimal set of rational support plans. The experiment of comparison with other MOGA set the iteration count to 375 and the number of plans to 350. Thus, it is critically important to have

as little time per iteration to build a diagram for a support plan as possible, even at the expense of diagram accuracy.

The different approach is implemented by E. Papadopoulou and D.T. Lee (EP&DTL) [20]. The algorithm handles the rectangular sites without additional division into points and uses the L_∞ distance metric. The sweep line algorithm processes events using each vertical edge of rectangles that allows constructing the Voronoi diagram with a larger number of rectangles and simplifies the adaptation of the algorithm for new events and structures. Thus, the present paper uses the EP&DTL algorithm as the core algorithm.

The EP&DTL algorithm finds the shortest critical areas on integrated circuits, such that which the plane of the diagram is connected everywhere. But the floor slabs can contain openings for different systems. Therefore, to correctly reflect the construction process, it is necessary to include a mechanism for processing rectangular holes in the Voronoi diagram.

3.2. Distance and bisector definitions. The EP&DTL algorithm uses the L_∞ distance instead of Euclidean, which simplifies Voronoi diagram calculation. Further, all the rules and conditions set in Section 2 use the L_∞ distance metric defined below:

$$d(p_1, p_2) = max(|x_1 - x_2|, |y_1 - y_2|), \forall p_1 = (x_1, y_1), p_2 = (x_2, y_2) \in Slab.$$

One of the main structures used in the EP&DTL algorithm is a bisector $\mathscr{b}v$. There are four general definitions that will be used further in the text. First of all, the main definition of the bisector as the locus of points at equal distance from two different edges:

$$\mathscr{b}v = Bv(R^{u1}, R^{u2}) = \{p: d(p, R^{u1}) = d(p, R^{u2})\}. \quad (3)$$

Secondly, the bisectors that represent edges of the elements:

$$\mathscr{b}v = Bv(R^u) = \{p: d(p, R^u) = 0\}. \quad (4)$$

Given a position of the vertical sweep line $\mathcal{L}$, the current position of bisectors $ep(\mathscr{b}v)$ (from equations (3), (4)) is equidistant from the sweep line:

$$ep(\mathscr{b}v) = p \in \mathscr{b}v: d(p, R^u) = \mathcal{L} - x_p, \forall R^u \in owners(\mathscr{b}v), \quad (5)$$

where $owners(\mathscr{b}v)$ are the edges given in the creation function of the bisector $\mathscr{b}v$.

The third type is a bisector that represents a single point:

$$\mathcal{bv} = Bv(R^{u1}, R^{u2}) = \{p: d(p, R^{u1}) = d(p, R^{u2}) = 0\}. \quad (6)$$

The current position of such bisector (6) is always the same point:

$$ep(\mathcal{bv}) = p \in \mathcal{bv}. \quad (7)$$

The last type is a bisector that depends on the edge of one element R^{u1} and lies on the edge of another element R^{u2}:

$$\mathcal{bv} = Bv(R^{u1}, R^{u2}) = \{p: d(p, R^{u2}) = 0\}. \quad (8)$$

The current position of this bisector from equation (8) is equidistant from the first owner and the sweep line position:

$$ep(\mathcal{bv}) = p \in \mathcal{bv}: d(p, R^{u1}) = \mathcal{L} - x_p, R^{u1} \in owners(\mathcal{bv}). \quad (9)$$

The creation function of the bisector types described further is set as one of the equations above or is defined additionally.

Conditions given in the bisector creation define the locus of points depending on the owners. Condition given by the current position defines a point depending on the owners and the sweep line position. The EP&DTL algorithm uses definitions of bisectors from (3) and (4) and the current position from (5), while others are used in the further modifications.

3.3. EP&DTL algorithm: the core algorithm definitions. Let us call bisectors induced by supports as real bisectors with denotation *bv*. The EP&DTL algorithm uses real bisectors equidistant from two support edges T_i^{u1} and T_i^{u2}, which can be defined by equation (3) as $bv = Bv(T_i^{u1}, T_i^{u2})$, and a support horizontal edge $T_i^u \in \{T_i^s, T_i^n\}$, which can be defined as a bisector by equation (4) as $bv = Bv(T_i^u)$.

A sweep line status $\mathcal{T}$ is a node of height-balanced binary tree consisting of a bisector:

$$\mathcal{T} = \langle bisector, left, right \rangle, \quad (10)$$

where *bisector*($\mathcal{T}$) is the bisector of $\mathcal{T}$; *left*($\mathcal{T}$), *right*($\mathcal{T}$) are the left and right nodes of $\mathcal{T}$.

The wavefront maintained by the tree contains segments between two consecutive nodes. If $bv = bisector(\mathcal{T})$, then the current position of the node $\mathcal{T}$ is denoted as $ep(\mathcal{T}) = ep(bv)$. The key of the node $\mathcal{T}$ in the tree equals the ordinate of its current position $key(\mathcal{T}) = y_{ep(bv)}$. Then *next*($\mathcal{T}$) and *prev*($\mathcal{T}$) are

the next and previous nodes of node $\mathcal{T}$ in the tree determined by *key*($\mathcal{T}$). This article implements a standard AVL tree for the sweep line status $\mathcal{T}$.

If owners are parallel in terms of coordinate, then the current position should be a point with the smaller maximum difference by another coordinate (Figure 1) that sets the unique current position. But there is an issue in the algorithm with west edge alignment; we discuss it later.

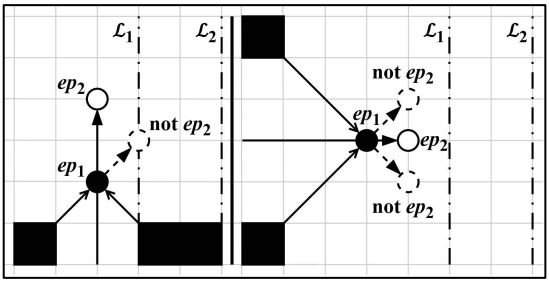


Fig. 1. Inexact current position with horizontal (left) and vertical (right) alignment

Finally, $\mathcal{Q}$ is a priority queue of diagram events. The EP&DTL algorithm defines three types of events: West Edge, East Edge and Spike. The type of some event $q \in \mathcal{Q}$ can be obtained by the function *type*(q):

$$type(q) = \{\text{West Edge, East Edge, Spike}\}. \tag{11}$$

West Edge and East Edge events occur as the sweep line reaches the edge of the site. Their priority equals the abscissa of the edge. West Edge event fixates part of the wavefront between intersection points, using bisectors induced from edge vertices, and creates two bisectors at these points and two bisectors at horizontal edges. East Edge event removes bisectors at horizontal edges and inserts two bisectors from edge vertices.

Spike event q_{Sp} occurs when two bisectors bv_1 and bv_2 intersect at a point $p_{\text{Sp}} = bv_1 \cap bv_2$. The priority of the Spike event q_{Sp} in the original paper [20] is the position of the sweep line $\mathcal{L}$ that is equidistant from p_{Sp} with abscissa x_{Sp} and the common owner of the bisectors $T_{\text{Sp}} = owners(bv_1) \cap owners(bv_2)$:

$$Priority(q_{\text{Sp}}) = x_{\text{Sp}} + d(p_{\text{Sp}}, T_{\text{Sp}}). \tag{12}$$

The Spike event removes bv_1 and bv_2 and creates a new bisector at p_{Sp}. Both bisectors must still be in $\mathcal{T}$ or the event should be ignored, which can be done by keeping a flag of removal state for each node $\mathcal{T}$.

The uniqueness of the current position from Figure 1 occurs differently for west and east vertical alignment. East alignment can be processed in the Spike event of east bisectors, with a new horizontal bisector. For the case of west edges, the condition is more complicated, especially with distortions by holes. One of the ways to handle the problem is to build the Voronoi diagram twice, flipping the abscissa of the slab elements, collecting edges from both diagrams and connecting disjoint Voronoi vertices. Section 5.2 considers this method and its limitations, while an applicable approach is a subject of future research. Figure 2 illustrates the problem of Voronoi edges' asymmetry.

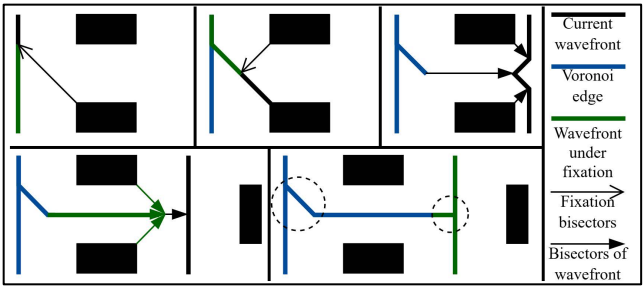


Fig. 2. Asymmetry in Voronoi diagram by edges alignment (shown by circles)

3.4. Slab boundary condition. Processing of slab boundaries during the sweep line algorithm in the L_∞ distance metric needs some modifications.

Definition 2. A bisector representing the slab horizontal edge $Slab^u \in \{Slab^s, Slab^n\}$ is called a border bisector and is defined by equation (4) as $bv_{\text{Brd}} = Bv(Slab^u)$. By equation (5) its current position follows the sweep line.

Definition 3. A bisector representing the slab west vertex of a horizontal edge $Slab^u \in \{Slab^s, Slab^n\}$ is called a turn bisector and is defined by equation (6) as $bv_{\text{Trn}} = Bv(Slab^w, Slab^u)$.

The real bisector bv induced by support edge T_i^{u1} and lying on the slab edge $Slab^{u2}$ is defined by equation (8) as $bv = Bv(T_i^{u1}, Slab^{u2})$. Then its current position is computed by equation (9) depending on T_i^{u1}.

Initially the sweep line status contains a pair of border bisectors for slab horizontal edges and a pair of turn bisectors at slab west vertices. Border bisectors always lie in the extreme positions in the AVL tree $\mathcal{T}$.

Let us denote a Turn event of intersection between a real bv and a turn bv_{Trn} as q_{Trn}, where x_{Trn} is the abscissa of the turn bisector position $ep(bv_{\text{Trn}})$. Then the priority of the Turn event is calculated as:

$$Priority(q_{\text{Trn}}) = x_{\text{Trn}} + d(ep(bv_{\text{Trn}}), owners(bv)).$$

The Turn event shown in Figure 3 removes *bv* and bv_{Trn} and inserts a real bisector on the slab horizontal edge $bv_{new} = Bv(T^{u1}, Slab^{u2})$, where $T^{u1} = owners(bv) \setminus owners(bv_{\text{Trn}})$, $Slab^{u2} = owners(bv_{\text{Trn}}) \setminus owners(bv)$.

Border and turn bisectors prevent violation of slab boundary. The accepted EP&DTL algorithm and proposed boundary process allow using Voronoi diagrams for simple building plans. The next section introduces holes processing in the Voronoi diagram to expand the scope of application.

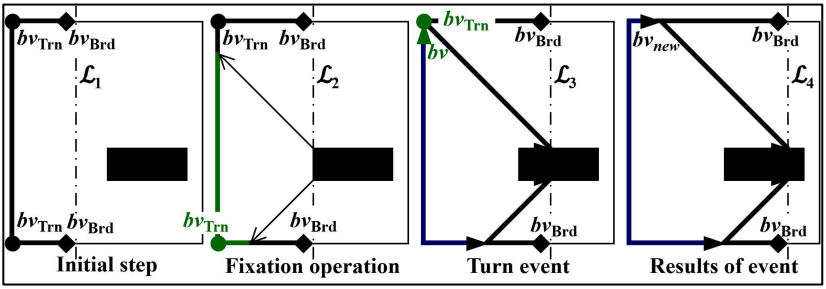


Fig. 3. Example of border and turn bisectors, Turn event (fixation shown in green)

4. Holes processing in Voronoi Diagram. Unlike supports, holes do not fixate wavefront segments but "shadow" them. Bisectors of a hole do not define edges in the Voronoi diagram, but instead determine shadow layers of other bisectors. Therefore, this section introduces new types of bisectors, a new structure of the wavefront with "shadows" and new types of events.

Let us define a real bisector *bv* with the owner T_i^{u1} on the hole edge $Hole^{u2}$ by equation (8) as $bv = Bv(T_i^{u1}, Hole^{u2})$.

Besides, let us define an additional real bisector *bv* with support edge owner T_i^u that is directed from the vertex v_k of $Hole_k$ by a set of conditions:

$$\begin{aligned} bv = Bv(T_i^u, v_k) = \{p: d(p, T_i^u) = d(v_k, T_i^u) + d(p, v_k); \\ \text{if } x_v = x_k \text{ then } x_p \neq x_v, y_p \neq y_v; \\ \text{else } sign(y_v - y_p) = sign(y_v - y_k - \frac{l_k}{2})\Big\}. \end{aligned} \tag{13}$$

The current position of the additional real bisector is calculated by equation (5) with T_i^u being the only owner. Such additional bisectors occur in some cases when a real bisector lying on the hole edge reaches its vertex, maintaining distractions from holes.

4.1. Virtual Bisectors in the Voronoi Wavefront. Let us call bisectors induced by holes virtual bisectors βv. Virtual bisectors determine the tree layer of other bisectors. The first virtual bisector owner is always a hole. There are four types of virtual bisectors with different behaviors.

Definition 4. A virtual bisector, which represents the hole horizontal edge $Hole_k^u \in \{Hole_k^s, Hole_k^n\}$, is called an edge virtual bisector and is defined by equation (4) as $\beta v_{\text{Edge}} = \mathcal{B}v_{\text{Edge}}(Hole_k^u) = Bv(Hole_k^u)$, where $\mathcal{B}v_{\text{Edge}}$ denotes the edge virtual bisector creation function.

Definition 5. A virtual bisector at hole vertex with horizontal edge $Hole_k^{u1} \in \{Hole_k^s, Hole_k^n\}$ and vertical edge $Hole_k^{u2} \in \{Hole_k^w, Hole_k^e\}$ is called a corner virtual bisector and is defined by equation (6) as $\beta v_{\text{Corner}} = \mathcal{B}v_{\text{Corner}}(Hole_k^{u1}, Hole_k^{u2}) = Bv(Hole_k^{u1}, Hole_k^{u2})$, where $\mathcal{B}v_{\text{Corner}}$ denotes the corner virtual bisector creation function.

Definition 6. A virtual bisector induced with hole adjacent edges starting from point $p_{\text{In}} \in Slab$ is called an inward virtual bisector:

$$\begin{gathered}\mathcal{B}v_{\text{In}}(Hole_k^{u1}, Hole^{u2}, R^{u3}, p_{\text{In}}) = \\ \{p: d(p, Hole_k^{u1}) = d(p, Hole^{u2})\},\end{gathered} \tag{14}$$

where $Hole_k^{u1} \in \{Hole_k^w, Hole_k^e\}$ is the hole vertical edge; if $Hole_k^{u1}$ is a west edge then $Hole^{u2} \in \{Hole_k^s, Hole_k^n\}$ is the hole horizontal edge; else $Hole^{u2} \in \{Hole^{+s}, Hole^{+n}\}$ is an extension of the horizontal edge from the vertex: $Hole^{+s} = ((x_k + l_k, y_k), (x_k + 2 \cdot l_k, y_k))$, $Hole^{+n} = ((x_k, y_k + w_k), ((x_k, y_k + 2 \cdot w_k))$; $R^{u3} \in R' \cup \{\varnothing\}$ is the owner of the wavefront segment in p_{In} if it exists. The owner R^{u3} affects only the definition of the wavefront segment owner, while p_{In} affects only the current position.

Inward virtual bisectors can be passive βv_{InP} or active βv_{InA} depending on the wavefront owner R^{u3} in p_{In}. For a passive inward virtual bisector βv_{InP} the third owner R^{u3} can be a slab, a hole edge or none. The current position of a passive inward virtual bisector βv_{InP} always equals p_{In}:

$$ep(\beta v_{\text{InA}}) = p_{\text{In}}. \tag{15}$$

Active inward virtual bisector βv_{InA} moves to the vertex of the hole and its third owner R^{u3} can be a support edge owner or none. The current position of an active inward virtual bisector βv_{InA} is obtained by a complex equality condition with an artificial sweep line $\mathcal{L}_{\text{Art}}$:

$$\begin{gathered}ep(\beta v_{\text{InA}}) = p \in \beta v_{\text{InA}}: \mathrm{d}(p, Hole^u) = \\ \mathcal{L}_{\text{Art}} - x_p, \forall Hole^u \in owners(\beta v_{\text{InA}}),\end{gathered} \tag{16}$$

where $\mathcal{L}_{\text{Art}} = \mathcal{L} - \mathcal{L}_{\text{In}} + 2 \cdot x_{\text{In}} - x_v$; $\mathcal{L}_{\text{In}}$ is the position of $\mathcal{L}$ at the moment of βv_{InA} creation; x_{In} is the abscissa of p_{In}; $x_v = Hole_k^{u1} \cap Hole^{u2}$ is the abscissa of the vertex.

Let us denote the creating function for a passive inward bisector as $\mathcal{B}v_{\text{InP}}$ and for an active one as $\mathcal{B}v_{\text{InA}}$; both are defined by equation (14), but create different types of inward virtual bisectors.

Definition 7. A virtual bisector lying parallel to the hole horizontal edge $Hole_k^{u1} \in \{Hole_k^s, Hole_k^n\}$ is called an outside virtual bisector. The behavior of an outside virtual bisector depends on its position: bounded by another hole or slab edge, or free. In the first case the outside virtual bisector is called limited and is defined by equation (8) as $\beta v_{\text{OutL}} = \mathcal{B}v_{\text{OutL}}(Hole_k^{u1}, R^{u2}) = Bv(Hole_k^{u1}, R^{u2})$, where $\mathcal{B}v_{\text{OutL}}$ is the denotation of the outside limited virtual bisector creation function, and R^{u2} is the edge of another hole or slab.

In the second case, when the position is free, the outside virtual bisector is called wide and is defined by equation (3) as $\beta v_{\text{OutW}} = \mathcal{B}v_{\text{OutW}}(Hole_k^{u1}, R^{u2}) = Bv(Hole_k^{u1}, R^{u2})$, where $\mathcal{B}v_{\text{OutW}}$ is the denotation of the outside wide virtual bisector creation function, R^{u2} is the edge that owns the wavefront segment in the beginning point p_{Out}, a support or hole, or an artificial rotated copy of $Hole_k^{u1}$ denoted as $Hole*$ if there are two owners of the wavefront segment:

$$\begin{gathered} Hole*(Hole_k^{u1}, p_{\text{Out}}) = ((x*, y*), (x*, y* + s \cdot l_k)), \\ x* = 2 \cdot x_{\text{Out}} - x_k; y* = 2 \cdot y_{\text{Out}} - y_k; s = sgn(y_{\text{Out}} - y_k). \end{gathered} \tag{17}$$

where $(x_{\text{Out}}, y_{\text{Out}}) = p_{\text{Out}}$; (x_k, y_k) are the coordinates of $Hole_k$, l_k is the abscissa dimension of $Hole_k$, that contains edge $Hole_k^{u1}$: $Hole_k^{u1} \in Hole_k$.

Table 2 summarizes the bisectors notation. Section 4.3 sets selection rules between specific creation functions. Once a virtual bisector is created it does not change its type. New inward or outside virtual bisector can be recreated with a new subtype and position.

Table 2. Notations for bisector creation and current position functions

Creation	Description	Position	Eq.
Real bisectors – induced by supports, set edges of the Voronoi diagram during fixation			
$Bv(T^{u1}, T^{u2})$	Bisector between edges of different supports	(5)	(3)
$Bv(T_i^{u1}, T_i^{u2})$	Bisector between edges of one support		
$Bv(T_i^u)$	Bisector along support horizontal edge		(4)
$Bv(T_i^u, p)$	Additional bisector from hole vertex		(13)
$Bv(T_i^{u1}, Slab^{u2})$	Bisector along slab edge	(9)	(8)
$Bv(T_i^{u1}, Hole^{u2})$	Bisector along hole edge		
Border and Turn bisectors – maintain slab boundary during intersection and fixation			
$Bv(Slab^u)$	Border bisector along slab horizontal edge	(5)	(4)
$Bv(Slab^w, Slab^{u2})$	Turn bisector at the slab west vertex	(7)	(6)

Creation	Description	Position	Eq.
Virtual bisectors – induced by holes, maintain shadows from the holes during intersection and fixation			
$\mathcal{B}v_{\text{Edge}}(Hole_k^{u})$	Edge bisector along hole horizontal edge	(5)	(4)
$\mathcal{B}v_{\text{Corner}}(Hole_k^{u1}, Hole_k^{u2})$	Corner bisector at the hole vertex	(7)	(6)
$\mathcal{B}v_{\text{InP}}(Hole_k^{u1}, Hole^{u2}, Hole^{u3}, p_{\text{In}})$	Passive inward bisector on some hole edge	(15)	(14)
$\mathcal{B}v_{\text{InP}}(Hole_k^{u1}, Hole^{u2}, Slab^{u3}, p_{\text{In}})$	Passive inward bisector on some slab edge		
$\mathcal{B}v_{\text{InA}}(Hole_k^{u1}, Hole^{u2}, T^{u3}, p_{\text{In}})$	Active inward bisector of support segment owner	(16)	
$\mathcal{B}v_{\text{In}}(Hole_k^{u1}, Hole^{u2}, \varnothing, p_{\text{In}})$	Passive or active inward bisector without segment owner depending on sweep line moment	(15)	
		(16)	
$\mathcal{B}v_{\text{OutL}}(Hole_k^{u1}, Hole^{u2})$	Outside limited bisector along hole edge	(9)	(8)
$\mathcal{B}v_{\text{OutL}}(Hole_k^{u1}, Slab^{u2})$	Outside limited bisector along slab edge		
$\mathcal{B}v_{\text{OutW}}(Hole_k^{u1}, T^{u2})$	Outside wide bisector with support segment owner	(5)	(3)
$\mathcal{B}v_{\text{OutW}}(Hole_k^{u1}, Hole^{u2})$	Outside wide bisector with hole segment owner		
$\mathcal{B}v_{\text{OutW}}(Hole_k^{u1}, Hole^{*})$	Outside wide bisector with two segment owners		

4.2. Shadows in the Voronoi Wavefront.

Definition 8. A shadow of the height-balanced binary tree is an inner equivalent tree between two nodes from the original tree called shadow nodes. Shadow node with virtual bisector βv is defined by:

$$\mathcal{T}^{\text{SH}} = \langle bisector, left, right, basis, end, dupl \rangle,$$

where $bisector(\mathcal{T}^{\text{SH}}) = \beta v$ is the virtual bisector of $\mathcal{T}^{\text{SH}}$; $left(\mathcal{T}^{\text{SH}})$, $right(\mathcal{T}^{\text{SH}})$ are the left and right nodes of $\mathcal{T}^{\text{SH}}$; $basis(\mathcal{T}^{\text{SH}})$ is the root node of the shadow; $end(\mathcal{T}^{\text{SH}})$ is the pair node on the opposite side of the shadow; $dupl(\mathcal{T}^{\text{SH}})$ is a node copy of $\mathcal{T}^{\text{SH}}$ inside the shadow which is called a duplicate node.

Definition 9. A duplicate node of some shadow node $\mathcal{T}^{\text{SH}}$ is:

$$\mathcal{T}^{\text{DP}} = \langle bisector, left, right, main \rangle,$$

where $main(\mathcal{T}^{\text{DP}}) = \mathcal{T}^{\text{SH}}$ is the shadow node of $\mathcal{T}^{\text{DP}}$; $bisector(\mathcal{T}^{\text{DP}}) = bisector(\mathcal{T}^{\text{SH}})$ is the virtual bisector of $\mathcal{T}^{\text{DP}}$, the same as $\mathcal{T}^{\text{SH}}$; $left(\mathcal{T}^{\text{DP}})$, $right(\mathcal{T}^{\text{DP}})$ are the left and right nodes of $\mathcal{T}^{\text{DP}}$, one of which is always $\varnothing$ because the duplicate node is the last node in the shadow.

Duplicate nodes ensure the exit of a bisector from the shadow through the intersection events.

The definitions of shadow and duplicate nodes extend the definition of real nodes from equation (10). Figure 4 shows an example of an AVL tree with a shadow layer in it.

Let us call nodes neither Shadow nor Duplicate Flat nodes $\mathcal{T}^{\text{F}}$. Shadow structure should provide a unique intersection between west-directed bisectors and the wavefront segments. Multiple intersection points

appear, because a hole should let the intersection be found on any of its edges or behind it. Therefore, a triple form of the hole shadow is proposed: upper and lower outer shadows and an inner shadow covered by them both, as shown in Figure 5. Inner shadow tracks the wavefront reaching the west edge of the hole. Outer shadows preserve the L_∞ structure of the wavefront and maintain a unique intersection.

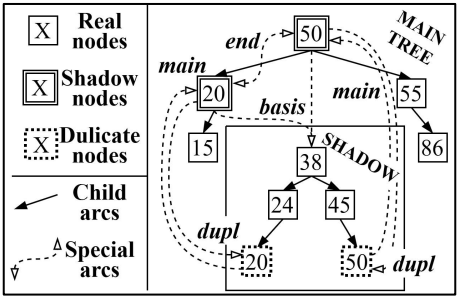


Fig. 4. An example of AVL tree with shadow

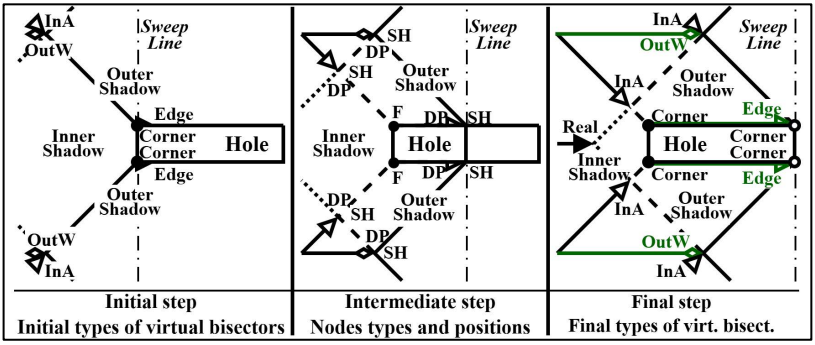


Fig. 5. Illustration of shadow structure; old shadow nodes are shown in green

Inward virtual bisectors enclose the inner shadow and, if active, move to the west vertices corner virtual bisectors. Outer shadow is active until the sweep line reaches the hole east edge, then it becomes inactive. In the active state, the outer shadow is enclosed between an outer virtual bisector and an edge virtual bisector. Inactivation replaces the outer virtual bisector with an inward virtual bisector directed to the hole east vertex where the corner virtual bisector lies.

The sweep line algorithm adds the inner shadow and the outer shadows when the sweep line reaches the west edge of a hole. To add a shadow, the algorithm should find the beginning positions of shadow nodes

and insert them, keeping nodes of the wavefront between them in the shadow using the basis node with duplicate nodes appended there. Shadows can contain each other entirely if both shadow nodes of another shadow are inside it, or partly if only one shadow node is covered.

Shadow nodes can be recreated separately retaining the same *end* and *basis*. The outer shadow is removed entirely when some real bisector reaches the east vertex of a hole. Inner shadow behaves differently: its shadow nodes can be removed, yet the shadow should still exist, which can be handled by a new type of nodes with real bisectors moving along the west edge of a hole.

Definition 10. A node with a real bisector *bv* lying on the west edge of the hole and being the last node in the inner shadow is called Blind Node:

$$\mathcal{T}^{\mathrm{BL}} = \langle bisector, left, right, basis, end \rangle,$$

where $bisector(\mathcal{T}^{\mathrm{BL}}) = bv$ is the real bisector of $\mathcal{T}^{\mathrm{BL}}$; $left(\mathcal{T}^{\mathrm{BL}})$, $right(\mathcal{T}^{\mathrm{BL}})$ are the left and right nodes of $\mathcal{T}^{\mathrm{BL}}$ one of which is always $\varnothing$ because the blind node is the last node of the shadow; $basis(\mathcal{T}^{\mathrm{BL}})$ is the root node of the shadow; $end(\mathcal{T}^{\mathrm{BL}})$ is the pair node on the opposite side of the shadow. Thus, the *end* function can set a paired shadow node outside of the shadow or a blind node inside it.

Figure 6 shows an example of the diagram and AVL tree structure when some inner shadow node is removed and a new blind node appears.

If the algorithm removes a blind node, it creates a new blind node at the same point or recreates the last node in the shadow as a new blind node with the same *end* and *basis*. When two blind nodes intersect, the algorithm removes the inner shadow entirely.

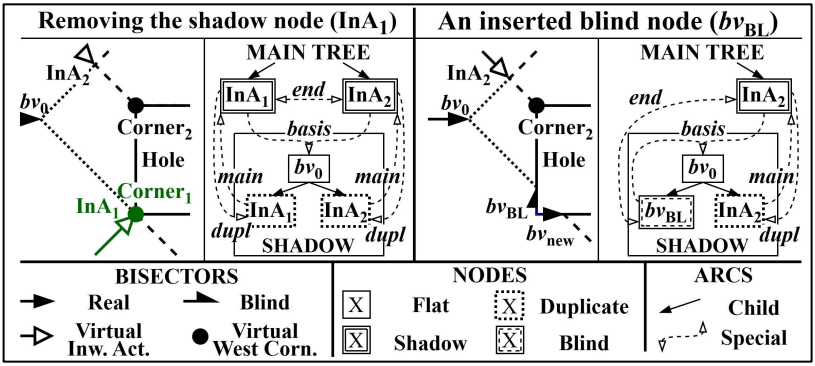


Fig. 6. Illustration of a blind node behavior in the diagram and the tree

4.3. Operations on the Wavefront. Let us determine the type operator of nodes such as $\mathcal{T}$ and any, real or virtual, bisectors such as $\mathscr{b}v$ by the function *type*:

1) Nodes: type($\mathcal{T}$) ∈ {F – flat, SH – shadow, DP – duplicate, BL – blind};

2) Bisectors: type($\mathscr{b}$v) ∈ {R – real, BD – border, TR – turn, VE – virtual edge, VC – virtual corner, VI – virtual inward, VO – virtual outside}.

There are some corollaries of the shadow structure and blind nodes:

- **if** $main(\mathcal{T}^{\mathrm{DP}}) = \mathcal{T}^{\mathrm{SH}}$ **then** $dupl(\mathcal{T}^{\mathrm{SH}}) = \mathcal{T}^{\mathrm{DP}}$,
- **if** $dupl(\mathcal{T}^{\mathrm{SH}}) = \mathcal{T}^{\mathrm{DP}}$ **then** $\mathcal{T}^{\mathrm{DP}} \in basis(\mathcal{T}^{\mathrm{SH}})$,
- **if** $type(\mathcal{T}_1) \in \{\mathrm{SH}, \mathrm{BL}\}$ **and** $type(\mathcal{T}_2) \in \{\mathrm{SH}, \mathrm{BL}\}$:
  - **if** $end(\mathcal{T}_2{}^{\mathrm{SH\ or\ BL}}) = \mathcal{T}_1$ **then** $end(\mathcal{T}_1{}^{\mathrm{SH\ or\ BL}}) = \mathcal{T}_2$,
  - **if** $end(\mathcal{T}_1{}^{\mathrm{SH\ or\ BL}}) = \mathcal{T}_2$ **then** $basis(\mathcal{T}_1{}^{\mathrm{SH\ or\ BL}}) = basis(\mathcal{T}_2{}^{\mathrm{SH\ or\ BL}})$,
  - **if** $end(\mathcal{T}_1{}^{\mathrm{SH\ or\ BL}}) = \mathcal{T}_2$ **and** $key(\mathcal{T}_1) < key(\mathcal{T}_2)$ **then**:

$$\forall \mathcal{T} \in basis(\mathcal{T}_1): key(\mathcal{T}_1{}^{\mathrm{DP\ or\ BL}}) \le key(\mathcal{T}) \le key(\mathcal{T}_2{}^{\mathrm{DP\ or\ BL}}).$$

Shadow and blind nodes should track changes on the basis. In practice, it is useful to have a reference to the basis with handlers of changing events. Besides, to avoid significant time consumption when computing distances with the holes condition (2) in practice, it is useful to keep a distance value of the bisector beginning point using the distance value of a former bisector or by a value itself.

Algorithm 1 defines the intersection of bisector $\mathscr{b}v'$ and node $\mathcal{T}$ with an inclusion flag $f \in \{-1, 1\}$ as *Intersect*($\mathscr{b}v'$, $\mathcal{T}$, f). If nodes lie on bisector $\mathscr{b}v'$, then the function finds the node with the lowest or highest abscissa, depending on $f = 1$ or $f = -1$. If all such nodes are at the same point, then the function returns the leftmost or rightmost node depending on f.

The function *Between* seeks the set of nodes in the tree $\mathcal{T}$ between two points p_{low} and $p_{\mathrm{up}} \in Slab$, such that $y_{\mathrm{up}} > y_{\mathrm{low}}$, as defined:

$$Between(\mathcal{T}, p_{\mathrm{up}}, p_{\mathrm{low}}) = \{\mathcal{T}' \in \mathcal{T} | y_{\mathrm{low}} \le key(\mathcal{T}') \le y_{\mathrm{up}}\}. \quad (18)$$

The operation searches for nodes whose keys are between the given points and goes through the tree using the *left* and *right* functions until conditions are met with O(log n + k), where n is the count of nodes in $\mathcal{T}$ and k is the output size.

Intersect($\mathcal{B}v'$, $\mathcal{T} \neq \varnothing$, $f \in \{-1, 1\}$)
1. **if** *ep*($\mathcal{T}$) lower $\mathcal{B}v'$ **and** *ep*(*next*($\mathcal{T}$)) lower $\mathcal{B}v'$ **then**
2. **return** ***Intersect***($\mathcal{B}v'$, *right*($\mathcal{T}$), *f*) ▷ seek intersection upper
3. **if** *ep*($\mathcal{T}$) upper $\mathcal{B}v'$ **and** *ep*(*next*($\mathcal{T}$)) upper $\mathcal{B}v'$ **then**
4. **return** ***Intersect***($\mathcal{B}v'$, *left*($\mathcal{T}$) , *f*) ▷ seek intersection lower
5. **if** *ep*($\mathcal{T}$) lower $\mathcal{B}v'$ **and** *ep*(*next*($\mathcal{T}$)) upper $\mathcal{B}v'$ **then**
6. **return** $\mathcal{B}v' \cap$ (*ep*($\mathcal{T}$), *ep*(*next*($\mathcal{T}$))) ▷ intersection point found
7. **if** *ep*($\mathcal{T}$) $\in \mathcal{B}v'$ **then** ▷ one or both lie on bisector
8. $\mathcal{T}_{\text{base}} \leftarrow \mathcal{T}$, $\mathcal{T}^{\text{on}} \leftarrow \{\mathcal{T}\}$, $p_{\text{base}} \leftarrow ep(\mathcal{T})$, $\mathcal{T}' \leftarrow prev(\mathcal{T})$
9. **else** $\mathcal{T}_{\text{base}} \leftarrow next(\mathcal{T})$, $\mathcal{T}^{\text{on}} \leftarrow \{next(\mathcal{T})\}$, $p_{\text{base}} \leftarrow ep(next(\mathcal{T}))$, $\mathcal{T}' \leftarrow \mathcal{T}$
10. add to $\mathcal{T}^{\text{on}}$ all nodes from left to right that lie on $\mathcal{B}v'$ by *left* and *right* from $\mathcal{T}_{\text{base}}$
11. $\mathcal{T}_{\text{Last}} \leftarrow \mathcal{T}^{\text{on}}$.Last(), $\mathcal{T}_{\text{First}} \leftarrow \mathcal{T}^{\text{on}}$.First(), $p_{\text{Last}} \leftarrow ep(\mathcal{T}_{\text{Last}})$, $p_{\text{First}} \leftarrow ep(\mathcal{T}_{\text{First}})$
12. **if** ($\exists \mathcal{T}^* \in \mathcal{T}^{\text{on}}$: $ep(\mathcal{T}^*) \neq p_{\text{base}}$ **and** sgn($x_{\text{Last}} - x_{\text{First}}$) = *f*) **or** ($x_{\text{base}} + 1$, $y_{\text{base}} - f$) $\in \mathcal{B}v'$ **then**
13. **return** $\mathcal{T}_{\text{Last}}$ ▷ return the rightest node
14. **else return** $\mathcal{T}_{\text{First}}$ ▷ return the most left node

Algorithm 1. Intersection operation

To create some types of bisectors we need to know an owner of the wavefront segment: real bisectors with two support owners; real bisectors on a hole or slab edge; additional real bisectors; outside virtual bisectors; inward virtual bisectors. The wavefront segment inside $\mathcal{T}$ at point p with a choice flag $f \in \{-1, 1\}$ is *Segment*($\mathcal{T}$, *p*, *f*):

$$Segment(\mathcal{T}, p) = \{(\mathcal{T}_1', \mathcal{T}_2') | key(\mathcal{T}_1') \le y_p \le key(\mathcal{T}_2'); \mathcal{T}_1', \mathcal{T}_2' \in \mathcal{T}, \mathcal{T}_2' = next(\mathcal{T}_1'), f = -1 \textbf{ and } ep(\mathcal{T}_1') \neq p \textbf{ or } f = 1 \textbf{ and } ep(\mathcal{T}_2') \neq p\}. \quad (19)$$

The function *Owner*($\mathcal{T}$, *p*, *f*) gets owner of the wavefront segment:

$$Owner(\mathcal{T}, p, f) = \{R^u | R^u \in owners(\mathcal{T}_1') \cap owners(\mathcal{T}_2'); \mathcal{T}_1', \mathcal{T}_2' \in \mathcal{T}, (\mathcal{T}_1', \mathcal{T}_2') = Segment(\mathcal{T}, p, f)\}. \quad (20)$$

The function *Owner* most of the time gives one owner, except for:

- if p lies between inward virtual bisectors inside the inner shadow, then the segment has two owners: the west edge of the hole and the same third owner of the inward bisectors;
- if p lies between an inward virtual bisector and a west corner virtual bisector of the same hole, then the segment has two owners: the west and horizontal edges of the hole;

- if p lies between a passive inward bisector without a third owner and a west corner virtual bisector of a different hole, then the segment has zero owners;
- if p lies between an inward virtual bisector and an outside virtual bisector of the same hole, then the segment has two owners: the horizontal edge of the hole and the third owner of the inward virtual bisector equal to the second owner of the outside bisector;
- if p lies between an active inward bisector with a support owner and a real bisector at the edge of the same hole, then the segment has two owners: the horizontal edge of the hole and the support edge owner.

In some other cases, there can be two or zero owners of the segment, yet such segments should never hold a new bisector and should not allow an intersection to pass through. The first case above can happen only for real bisector creation, and the second and third cases only for virtual creation.

Let us define the creation functions of bisectors at a point p on the wavefront with node $\mathcal{T}$ with direction flag $f \in \{-1, 1\}$ by rules for each bisector type. Because the function *Segment* has $O(\log n)$ time complexity, therefore the function *Owner* and creation functions have $O(\log n)$ time complexity as well.

With the function *Owner* returning one owner, except for the first case above, the function $Create_{\mathrm{R}}(\mathcal{T}, T_i^u, p, f)$ creates a real bisector:

- **if** $|Owner(\mathcal{T}, p, f)| = 1$ **then return** $Bv(T_i^u, Owner(\mathcal{T}, p, f))$;
- **else** $T_j^{u2} \in Owner(\mathcal{T}, p, f)$, $T_j \in T$: **return** $Bv(T_i^{u1}, T_j^{u2})$.

For the inward virtual bisector two owners are given. The function $Create_{\mathrm{In}}(\mathcal{T}, Hole_k^{u1}, Hole^{u2}, p_{\mathrm{In}}, f)$ is defined:

- **if** $|Owner(\mathcal{T}, p_{\mathrm{In}}, f)| = 1$ **and** $T_i^{u3} \in Owner(\mathcal{T}, p_{\mathrm{In}}, f)$: $T_i \in T$ **then return** $\mathcal{B}v_{\mathrm{InA}}(Hole_k^{u1}, Hole^{u2}, T_i^{u3}, p_{\mathrm{In}})$;
- **else if** $|Owner(\mathcal{T}, p_{\mathrm{In}}, f)| = 1$ **and** $R^u \in Owner(\mathcal{T}, p_{\mathrm{In}}, f)$: $p_{\mathrm{In}} \in R^u$ **then return** $\mathcal{B}v_{\mathrm{InP}}(Hole_k^{u1}, Hole^{u2}, R^u, p_{\mathrm{In}})$;
- **else if** $|Owner(\mathcal{T}, p_{\mathrm{In}}, f)| \neq 0$ **and** $Hole_k^u \in Owner(\mathcal{T}, p_{\mathrm{In}}, f)$: $\mathcal{L} \geq x_k + l_k$ **then return** $\mathcal{B}v_{\mathrm{InA}}(Hole_k^{u1}, Hole^{u2}, \varnothing, p_{\mathrm{In}})$;
- **return** $\mathcal{B}v_{\mathrm{InP}}(Hole_k^{u1}, Hole^{u2}, \varnothing, p_{\mathrm{In}})$.

For the outside virtual bisector, one owner is given, while the second is the wavefront segment owner. If there are two or zero owners, then we need to build an artificial edge owner by equation (17). The function $Create_{\mathrm{Out}}(\mathcal{T}, Hole_k^u, p, f)$ with the inducing owner $Hole_k^u$ is defined as:

- **if** $|Owner(\mathcal{T}, p, f)| \neq 1$ **then return** $\mathcal{B}v_{\mathrm{OutW}}(Hole_k^u, Hole^*(Hole_k^u, p))$;
- **else if** $R^u \in Owner(\mathcal{T}, p, f)$: $p \in R^u$ **then return** $\mathcal{B}v_{\mathrm{OutL}}(Hole_k^u, R^u)$;
- **else return** $\mathcal{B}v_{\mathrm{OutW}}(Hole_k^u, Owner(\mathcal{T}, p, f))$.

The function $Form(\mathcal{T}^{SH})$ returns a form of the shadow by a shadow node $\mathcal{T}^{SH}$: if $type(bisector(\mathcal{T}^{SH})) \in \{VE, VC\}$ or $type(bisector(end(\mathcal{T}^{SH}))) \in \{VE, VC\}$, then $Form(\mathcal{T}^{SH}) = Outer$; else $Form(\mathcal{T}^{SH}) = Inner$.

The function $Insert_t(\mathcal{T}, \mathcal{bv})$ adds bisector $\mathcal{bv}$ to the tree $\mathcal{T}$ with node type $t \in \{F, SH, DP, BL\}$. If the function inserts a shadow node, then it inserts a duplicate to the shadow of it as well. The function $Insert(\mathcal{T}, \beta v_1, \beta v_2)$ adds a shadow from the virtual bisectors βv_1 and βv_2 into tree $\mathcal{T}$ keeping nodes between them and their duplicates into the shadow. When inserting a shadow node $\mathcal{T}^{SH}$ of an inner shadow, we bond it with its pair using $Inner(Hole_k, f) = \mathcal{T}^{SH}$, $f = -1$ or 1 for the lower or upper node; if the wavefront reaches the west corners of $Hole_k$, then $Inner(Hole_k, f) = \varnothing$.

The function $Voronoi(T_i, (p_1, p_2))$: VE_i.Add((p_1, p_2)) fixates (p_1, p_2) for site T_i. Algorithm 2 describes the function $Voronoi(\mathcal{T}_{next}, \mathcal{T}_{prev}, T_{fix}, a, b)$ adding a Voronoi edge between nodes or a node and a point if a or $b \neq \varnothing$ by T_{fix}.

Voronoi($\mathcal{T}_{next}$, $\mathcal{T}_{prev}$: $\mathcal{T}_{next} = next(\mathcal{T}_{prev})$, $T_{fix} \in T$, $a \in Slab \cup \{\varnothing\}$, $b \in Slab \cup \{\varnothing\}$)
1. set p_{next} as a if it is given or as $ep(\mathcal{T}_{next})$ otherwise
2. set p_{prev} as b if it is given or as $ep(\mathcal{T}_{prev})$ otherwise
3. **if** $T_{mid}^{u} \in owners(\mathcal{T}_{prev}) \cap owners(\mathcal{T}_{next})$: $T_{mid} \in T$ **and** $T_{mid} \neq T_{fix}$ **then**
4. ***Voronoi***(T_{mid}, (p_{next}, p_{prev})), ***Voronoi***(T_{fix}, (p_{next}, p_{prev}))
5. **if** $R_{mid}^{u} \in owners(\mathcal{T}_{prev}) \cap owners(\mathcal{T}_{next})$: $p_{prev} \in R_{mid}^{u}$ **and** $p_{next} \in R_{mid}^{u}$ **and** $R_{mid} \notin T$ **then**
6. ***Voronoi***(T_{fix}, (p_{next}, p_{prev}))

Algorithm 2. Adding a Voronoi edge between two nodes

The function $Fixate(\mathcal{bv}')$ fixates a Voronoi edge by a real bisector $\mathcal{bv}'$ with two different owners. The function creates a Voronoi edge for every support owner from the beginning point to the current position of the bisector $ep(\mathcal{bv}')$. The function $Remove(\mathcal{T}, \mathcal{T}')$ deletes $\mathcal{T}'$ from $\mathcal{T}$ and fixates it: $\mathcal{T}$.Remove($\mathcal{T}'$), $Fixate(bisector(\mathcal{T}'))$.

The function $\mathcal{B}v_{insd}(\mathcal{T}^{SH})$ sets an intersection bisector of $\mathcal{T}^{SH}$, outside or inward. If $Hole_k \in Hole$ is the hole of $\mathcal{T}^{SH}$ and $end(\mathcal{T}^{SH})$, $p_{End} = ep(end(\mathcal{T}^{SH}))$, $type = type(bisector(\mathcal{T}^{SH})) \in \{VI, VO\}$, then:

- **if** $type = VI$ **then return** $\mathcal{B}v(owners(\mathcal{T}^{SH}))$ without third owner;
- **else return** $\mathcal{B}v(\{(p_{End}, (x_k + l_k, y_k)), (p_{End}, (x_{End}, y_k + w_k))\})$ – sets the bisector from the position of the paired shadow node.

The function $Front(\mathcal{T}_1)$ defines a root of the tree by searching for the leftmost node $\mathcal{T}_{min}$ from $\mathcal{T}_1$. If $type(\mathcal{T}_{min}) \in \{DP, BL\}$, then the function returns a basis from the shadow or blind node. Else it returns a root of the main tree $\mathcal{T}$.

Algorithm 3 describes the function $Relocate(\mathcal{T}, \mathcal{T}^{SH}, p)$ finding a new bisector for $\mathcal{T}^{SH}$ with an outside or inward bisector in position p of the tree $\mathcal{T}$.

Relocate($\mathcal{T} \neq \varnothing$, $\mathcal{T}^{SH}$: *type*(*bisector*($\mathcal{T}^{SH}$)) $\in$ {VI, VO}, $p \in$ *Slab*)
1. $f \leftarrow$ sgn(*key*($\mathcal{T}^{SH}$) – *key*(*end*($\mathcal{T}^{SH}$))), set $Hole_k^{u1}$ as the edge inducing the bisector of $\mathcal{T}^{SH}$
2. **if** *type*(*bisector*($\mathcal{T}^{SH}$)) = VI **then**
3. set $Hole^{u2}$ as the second edge inducing the bisector of $\mathcal{T}^{SH}$
4. **return** ***Create***$_{\mathbf{In}}$($\mathcal{T}$, $Hole_k^{u1}$, $Hole^{u2}$, p, f) ▷ return inward bisector
5. **else return** ***Create***$_{\mathbf{Out}}$($\mathcal{T}$, $Hole_k^{u1}$, p, f) ▷ return outside bisector

Algorithm 3. Relocation of outside or inward virtual bisector

The function *Inject*($\mathcal{T}$, $\mathcal{T}^{SH}$, p) inserts a relocated shadow node of the previous shadow node $\mathcal{T}^{SH}$ into the tree $\mathcal{T}$ at position p: $\mathcal{T}_{new}^{SH} = Insert_{SH}(\mathcal{T}, Relocate(\mathcal{T}, \mathcal{T}^{SH}, p))$, $end(\mathcal{T}_{new}^{SH}) = end(\mathcal{T}^{SH})$.

Algorithm 4 defines the function *Swap*($\mathcal{T}$, $\mathcal{T}_1^{DP}$, $\mathcal{T}_2^{SH}$) swapping shadow nodes *main*($\mathcal{T}_1^{DP}$) and $\mathcal{T}_2^{SH}$ with inward or outside bisectors in tree $\mathcal{T}$:

Swap($\mathcal{T} \neq \varnothing$, $\mathcal{T}_1^{DP}$, $\mathcal{T}_2^{SH} \in$ *basis*(*main*($\mathcal{T}_1^{DP}$)))
1. $\mathcal{T}_1^{SH} \leftarrow$ *main*($\mathcal{T}_1^{DP}$), $p_{1^*} \leftarrow$ ***Intersect***($\mathcal{B}v_{insd}$($\mathcal{T}_1^{SH}$), *basis*($\mathcal{T}_2^{SH}$), 1) ▷ new position of $\mathcal{T}_1^{SH}$
2. set (p_{Up}, p_{Low}) as (*ep*($\mathcal{T}_2^{SH}$), p_{1^*}) if *key*($\mathcal{T}_1^{SH}$) < *key*($\mathcal{T}_2^{SH}$) or vice versa otherwise
3. $\mathcal{T}^{btwn} \leftarrow$ ***Between***(*basis*($\mathcal{T}_2^{SH}$), p_{Up}, p_{Low})
4. $\beta v_{1^*} \leftarrow$ ***Relocate***(*basis*($\mathcal{T}_2^{SH}$), $\mathcal{T}_1^{SH}$, p_{1^*}),
5. $\beta v_{2^*} \leftarrow$ ***Relocate***($\mathcal{T}$, $\mathcal{T}_2^{SH}$, *ep*($\mathcal{T}_1^{DP}$))
6. ***Remove***(*basis*($\mathcal{T}_2^{SH}$), $\forall \mathcal{T}^* \in \mathcal{T}^{btwn}$)
7. ***Remove***(*basis*($\mathcal{T}_2^{SH}$), *dupl*($\mathcal{T}_2^{SH}$)), ***Remove***(*basis*($\mathcal{T}_1^{SH}$), $\mathcal{T}_2^{SH}$); ▷ remove $\mathcal{T}_2^{SH}$
8. ***Remove***(*basis*($\mathcal{T}_1^{SH}$), $\mathcal{T}_1^{DP}$), ***Remove***($\mathcal{T}$, $\mathcal{T}_1^{SH}$) ▷ remove $\mathcal{T}_1^{SH}$
9. $\mathcal{T}_{2^*}^{SH} \leftarrow$ ***Insert***$_{\mathbf{SH}}$($\mathcal{T}$, βv_{2^*}), *end*($\mathcal{T}_{2^*}^{SH}$) = *end*($\mathcal{T}_2^{SH}$) ▷ relocate $\mathcal{T}_2^{SH}$ to $\mathcal{T}$
10. $\mathcal{T}_{1^*}^{SH} \leftarrow$ ***Insert***$_{\mathbf{SH}}$(*basis*($\mathcal{T}_{2^*}^{SH}$), βv_{1^*}), *end*($\mathcal{T}_{1^*}^{SH}$) = *end*($\mathcal{T}_1^{SH}$) ▷ relocate $\mathcal{T}_1^{SH}$ into $\mathcal{T}_{2^*}^{SH}$
11. ***Insert***$_{\mathbf{type}(\mathcal{T}^*)}$(*basis*($\mathcal{T}_{1^*}^{SH}$), $\forall \mathcal{T}^* \in \mathcal{T}^{btwn}$) ▷ reinsert nodes of $\mathcal{T}^{btwn}$
12. **return** ($\mathcal{T}_{1^*}^{SH}$, $\mathcal{T}_{2^*}^{SH}$, $\mathcal{T}^{btwn}$) ▷ relocated nodes and $\mathcal{T}^{btwn}$

Algorithm 4. Swapping shadow nodes in tree

Algorithm 5 defines the function *OnShadow*($\mathcal{T}_{on}$, $f \in \{-1, 1\}$) giving a shadow node of the intersected shadow in the segment from node $\mathcal{T}_{on}$ in ordinate direction f and ascending abscissa. The time complexity of the function is $O(K)$, where K is the count of holes.

OnShadow($\mathcal{T}_{on} \neq \varnothing$: *type*(*bisector*($\mathcal{T}_{on}$)) $\in$ {VE, VC, VI ,VO}, $f \in \{-1, 1\}$)
1. $p_{on} \leftarrow$ *ep*($\mathcal{T}_{on}$), set $\mathcal{T}'$ as *next*($\mathcal{T}_{on}$) if f = 1 or as *prev*($\mathcal{T}_{on}$) otherwise, $p' \leftarrow$ *ep*($\mathcal{T}'$)
2. **while** ($x' > x_{on}$ **and** sgn($y' - y_{on}$) = f) **or** $p' = p_{on}$ **do** ▷ the same point or co-directional
3. check if *type*(*bisector*($\mathcal{T}_{on}$)) $\notin$ {VE, VC, VI ,VO} then **return** $\varnothing$ ▷ must be virtual
4. **if** *type*($\mathcal{T}'$) = SH **then** ▷ east corner of hole
5. **return** *end*($\mathcal{T}'^{SH}$);
6. **if** *type*(*bisector*($\mathcal{T}'$)) = VC **then** ▷ west corner of hole
7. set $Hole_k$ as hole owner of node $\mathcal{T}'$ with west corner bisector by *owners*($\mathcal{T}'$)
8. **return** ***Inner***($Hole_k$, f) ▷ inward bisector to the corner $\mathcal{T}'$
9. set $\mathcal{T}'$ as *main*($\mathcal{T}'^{DP}$) if *type*($\mathcal{T}'$) = DP ▷ get a shadow node if duplicate

10. set $\mathcal{T}'$ as *next*($\mathcal{T}'$) if $f = 1$ or as *prev*($\mathcal{T}'$) otherwise, $p' \leftarrow ep(\mathcal{T}')$
11. **return** ∅ ▷ no intersected shadow in segment

Algorithm 5. Checking existence of intersected shadow in wavefront segment

Algorithm 6 defines the function *Fixate*($\mathcal{T}$, bv_a' | a, bv_b' | b, $T_i \in T$, $f \in \{-1, 1\}$) fixating nodes between bisectors or points bv_a' or a and bv_b' or b in $\mathcal{T}$ by site T_i with an inclusion flag f. If $f = -1$ then the algorithm does not process shadow nodes at the intersection points; for the West Edge event, the initial state f equals –1. There are three cases processing shadows:

- if bv_a' or bv_b' intersects a pair of shadow nodes, and only one of the pair lies between, then we fixate part of their shadow and relocate the shadow node in the between set at a new position (Figure 7a);
- if there is a lower shadow node of an outer shadow between the intersection points and the upper shadow node is also in the between set, then the algorithm fixates the shadow of the shadow node entirely (Figure 7b);
- if there is a hole west vertex in the between set that is not on the slab edge then the algorithm fixates part of the inner shadow maintained by the inward shadow node directed to the related west vertex (Figure 7c).

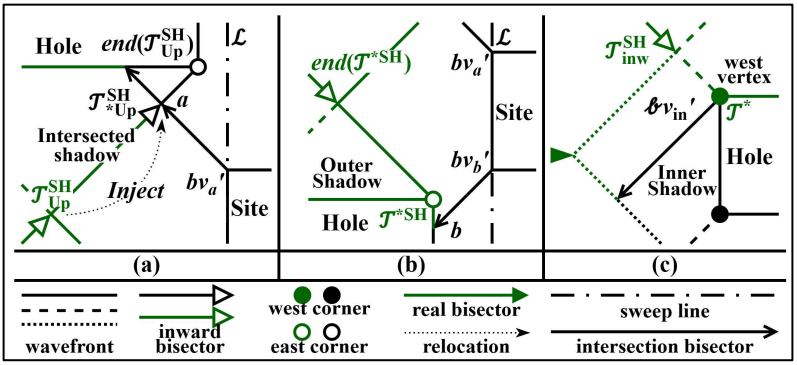


Fig. 7. Special cases of fixation: intersecting a shadow a) fixating an outer shadow, b) fixating an inner shadow, c) fixated and removed elements shown in green

Fixate($\mathcal{T} \neq \varnothing$, bv_a' **or** a, bv_b' **or** b, $T_i \in T$, $f \in \{-1, 1\}$)
1. $a \leftarrow$ ***Intersect***(bv_a', $\mathcal{T}$, f), $b \leftarrow$ ***Intersect***(bv_b', $\mathcal{T}$, f) ▷find intersection points if not given
2. $\mathcal{T}^{btwn} \leftarrow$ ***Between***($\mathcal{T}$, a, b), $\mathcal{T}_{First} \leftarrow \mathcal{T}^{btwn}$.First(), $\mathcal{T}_{Last} \leftarrow \mathcal{T}^{btwn}$.Last()
3. **if** $f = -1$ **then** ▷ is there shadow nodes at a or b
4. **while** $\mathcal{T}^{btwn} \neq \varnothing$ **and** *type*($\mathcal{T}_{First}$) = SH **do** ▷ shadow nodes in point b
5. $\mathcal{T}^{btwn}$.Remove($\mathcal{T}_{First}$), $\mathcal{T}_{First} \leftarrow \mathcal{T}^{btwn}$.First()
6. **while** $\mathcal{T}^{btwn} \neq \varnothing$ **and** *type*($\mathcal{T}_{Last}$) = SH **do** ▷ shadow nodes in point a
7. $\mathcal{T}^{btwn}$.Remove($\mathcal{T}_{Last}$), $\mathcal{T}_{Last} \leftarrow \mathcal{T}^{btwn}$.Last()

8. $\mathcal{T}_{\text{Low}}^{\text{SH}} \leftarrow$ ***OnShadow***$(\mathcal{T}_{\text{First}} \neq \varnothing, -1)$, $\mathcal{T}_{\text{Up}}^{\text{SH}} \leftarrow$ ***OnShadow***$(\mathcal{T}_{\text{Last}} \neq \varnothing, 1)$
9. **if** $\mathcal{T}_{\text{Up}}^{\text{SH}} \neq \varnothing$ **then** ▷ shadow intersected in point *a*
10. ***Fixate***$(basis(\mathcal{T}_{\text{Up}}^{\text{SH}}), bv_a', ep(\mathcal{T}_{\text{Up}}^{\text{SH}}), T_i, f)$ ▷ fixate under *a* (Fig. 7a);
11. **if** $\mathcal{T}^{\text{btwn}} = \varnothing$ **then** ▷ no nodes between *a* and *b* case
12. $(\mathcal{T}_{\text{First}}, \mathcal{T}_{\text{Last}}) \leftarrow$ ***Segment***$(\mathcal{T}, b, 1)$, ***Voronoi***$(\mathcal{T}_{\text{First}}, \mathcal{T}_{\text{Last}}, T_i, a, b)$
13. **else** ***Voronoi***$(\mathcal{T}_{\text{First}}, prev(\mathcal{T}_{\text{First}}), T_i, \varnothing, b)$, ***Voronoi***$(next(\mathcal{T}_{\text{Last}}), \mathcal{T}_{\text{Last}}, T_i, a, \varnothing)$
14. **for** $\mathcal{T}^* \in \mathcal{T}^{\text{btwn}}$ **do** ▷ nodes in $\mathcal{T}^{\text{btwn}}$ from upper to lower
15. $\mathscr{b}v^* \leftarrow bisector(\mathcal{T}^*)$, $p^* \leftarrow ep(\mathscr{b}v^*)$, $p_{\text{next}} \leftarrow ep(bisector(next(\mathcal{T}^*)))$
16. **if** $type(\mathcal{T}^*)$ = SH **and** ***Form***$(\mathcal{T}^{*\text{SH}})$ = Outer **and** $y^* < key(end(\mathcal{T}^{*\text{SH}}))$ **then** ▷ Fig. 7b
17. ***Fixate***$(basis(\mathcal{T}^{*\text{SH}}), p^*, ep(end(\mathcal{T}^{*\text{SH}})), T_i, -1)$)
18. **else if** $type(\mathscr{b}v^*)$ = VC **and** $type(\mathcal{T}^*)$ = F **and** $\nexists Slab^u$: $p^* \in Slab^u$ **then** ▷Fig. 7c
19. $(Hole_k^w, Hole_k^u) \leftarrow owners(\beta v^*)$ ▷ west and horizontal edge from βv^*
20. **if** $Hole_k^u = Hole_k^s$ **then** ▷ find extension of horizontal edge
21. $Hole^{+w} \leftarrow ((x_k, y_k), (x_k, y_k - w_k))$, $f_{\text{inw}} \leftarrow -1$
22. **else** $Hole^{+w} \leftarrow ((x_k, y_k + w_k), (x_k, y_k + 2 \cdot w_k))$, $f_{\text{inw}} \leftarrow 1$
23. $\mathcal{T}_{\text{inw}}^{\text{SH}} \leftarrow$ ***Inner***$(Hole_k, f_{\text{inw}})$, $p_{\text{inw}} \leftarrow ep(\mathcal{T}_{\text{inw}}^{\text{SH}})$, $\mathscr{b}v_{\text{in}}' \leftarrow$ ***Bv***$(Hole^{+w}, Hole_k^u)$
24. ***Fixate***$(basis(\mathcal{T}_{\text{inw}}^{\text{SH}}), p_{\text{inw}}, \mathscr{b}v_{\text{in}}', T_i, -1)$ ▷ fixate part of inner shadow
25. $\mathcal{T}_w^{\text{BL}} \leftarrow Insert_{\text{BL}}(basis(\mathcal{T}_{\text{inw}}^{\text{SH}}), Bv(T_i, Hole_k^w))$, $end(\mathcal{T}_w^{\text{BL}}) \leftarrow end(\mathcal{T}_{\text{inw}}^{\text{SH}})$
26. Insert in Q a new intersection event for $\mathcal{T}_w^{\text{BL}}$ if needed
27. add a Voronoi edge ***Voronoi***$(next(\mathcal{T}^*), \mathcal{T}^*, T_i, \varnothing, \varnothing)$ if $\mathcal{T}^* \neq \mathcal{T}_{\text{Last}}$
28. ***Remove***$(\mathcal{T}, \forall \mathcal{T}^* \in \mathcal{T}^{\text{btwn}})$
29. **if** $\mathcal{T}_{\text{Low}}^{\text{SH}} \neq \varnothing$ **then** ▷ shadow intersected in point *b*
30. ***Fixate***$(basis(\mathcal{T}_{\text{Low}}^{\text{SH}}), ep(\mathcal{T}_{\text{Low}}^{\text{SH}}), bv_b', T_i, f)$ ▷ fixate above *b* (Fig. 7a)
31. $\mathcal{T}_{\text{Low}^*}^{\text{SH}} \leftarrow$ ***Inject***$(\mathcal{T}, \mathcal{T}_{\text{Low}}^{\text{SH}}, b)$ ▷ relocate lower shadow node
32. **else** $\mathcal{T}_b \leftarrow$ ***Insert***$_{\text{F}}(\mathcal{T}$, ***Create***$_{\text{R}}(\mathcal{T}, T_i^s, b, -1))$ ▷ create bisector in lower point
33. **if** $\mathcal{T}_{\text{Up}}^{\text{SH}} = \varnothing$ **then**
34. $\mathcal{T}_a \leftarrow$ ***Insert***$_{\text{F}}(\mathcal{T}$, ***Create***$_{\text{R}}(\mathcal{T}, T_i^n, a, 1))$ ▷ create bisector in upper point
35. **else** $\mathcal{T}_{\text{Up}^*}^{\text{SH}} \leftarrow$ ***Inject***$(\mathcal{T}, \mathcal{T}_{\text{Up}}^{\text{SH}}, a)$ ▷ relocate upper shadow node
36. Insert in Q new intersection events for $\mathcal{T}_a$ and $\mathcal{T}_b$ or for $\mathcal{T}_{\text{Up}^*}^{\text{SH}}$ and $\mathcal{T}_{\text{Low}^*}^{\text{SH}}$ if needed

Algorithm 6. Fixation of the nodes between two points

The worst case of the count of *Fixate* executions follows. The left half of the holes lies lower than the other half in a descending order of abscissa and ordinate with upper outer shadows and upper shadow nodes of inner shadows enveloping each other, while the right half lies in the same order; therefore, their lower shadow nodes of inner shadows lie on the segment between the upper shadow node of inner shadow and the northwest corner of the uppermost hole from the left half. This is the worst possible case of a single *Fixate* execution, because the fixation from the northwest corner of every hole (step 18) from the right half can possibly interact with each upper outer shadow and upper parts of inner shadows of every hole from the left half. Then the *Fixate* operation will happen $K^2/2 + 5 \cdot K + 1$ times, where K is the count of holes. But the average number of occurrences for every outer shadow and twice for every inner shadow is $4 \cdot K + 1$ times.

The operation with the worst time complexity in each execution is step 28 removing nodes from the between set with $O(n \log n)$ in the worst case of each node removed (n is the count of nodes) and *OnShadow* execution with $O(K)$ in the worst case. Therefore, the time complexity of the *Fixate* operation is $O(K^2 \cdot n \cdot \log(n) + K^3)$.

Therefore, the proposed structure of the shadows in AVL trees handles the holes processing. Table 3 describes the wavefront functions and their time complexity, where n is the count of input nodes, k is the count of output nodes, and K is the count of holes. Further, the new sweep line events corresponding to the holes are described.

Table 3. Wavefront operations and functions

Function	Description	Time	Ref.
Intersect($\mathcal{B}v'$, $\mathcal{T}$, f)	Gets wavefront intersection by bisector	$O(\log n)$	Alg. 1
Between($\mathcal{T}$, p_{up}, p_{low})	Gets nodes in tree between points	$O(\log n + k)$	(18)
Segment($\mathcal{T}$, p, f)	Gets nodes of wavefront segment in point	$O(\log n)$	(19)
Owner($\mathcal{T}$, p, f)	Gets edge owner of wavefront in point	$O(\log n)$	(20)
Create$_{\text{R}}$($\mathcal{T}$, T_i^u, p, f)	Creates real bisector by owner in point	$O(\log n)$	–
Create$_{\text{In}}$($\mathcal{T}$, *Hole*$_k^{u1}$, *Hole*^{u2}, p_{In}, f)	Creates inward bisector by owners in point	$O(\log n)$	–
Create$_{\text{Out}}$($\mathcal{T}$, *Hole*$_k^u$, p, f)	Creates outside bisector by owner in point	$O(\log n)$	–
Form($\mathcal{T}^{\text{SH}}$)	Gets form of the shadow (Outer or Inner)	$O(1)$	–
Insert$_{type}$($\mathcal{T}$, $\mathcal{B}v$)	Creates a node of chosen type by bisector	$O(\log n)$	–
Insert($\mathcal{T}$, βv_1, βv_2)	Creates shadow nodes pair by bisectors	$O(\log n)$	–
Inner(*Hole*$_k$, f)	Gets a shadow node of inner shadow	$O(1)$	–
Voronoi(T_i, (p_1, p_2))	Sets Voronoi edge for support by segment	$O(1)$	–
Voronoi($\mathcal{T}_{\text{next}}$, $\mathcal{T}_{\text{prev}}$, T_i, a, b)	Sets Voronoi edge between nodes or points	$O(1)$	Alg. 2
Fixate($\mathcal{B}v'$)	Fixates Voronoi edge by bisector	$O(1)$	–
Remove($\mathcal{T}$, $\mathcal{T}'$)	Removes node from tree and fixates it	$O(\log n)$	–
$\mathcal{B}v_{\text{insd}}$($\mathcal{T}^{\text{SH}}$)	Gets intersection bisector by inward, outside	$O(1)$	–
Front($\mathcal{T}_1$)	Gets root node of the tree in which $\mathcal{T}_1$ is	$O(\log n)$	–
Relocate($\mathcal{T}$, $\mathcal{T}^{\text{SH}}$, p)	Relocates shadow node to point into tree	$O(\log n)$	Alg. 3
Inject($\mathcal{T}$, $\mathcal{T}^{\text{SH}}$, p)	Inserts a relocated shadow node	$O(\log n)$	–
Swap($\mathcal{T}$, $\mathcal{T}_1^{\text{DP}}$, $\mathcal{T}_2^{\text{SH}}$)	Swaps inward or outside shadow nodes	$O(n \log n)$	Alg. 4
OnShadow($\mathcal{T}_{\text{on}}$, f)	Checks if shadow under wavefront segment	$O(K)$	Alg. 5
Fixate($\mathcal{T}$, $bv_a' \mid a$, $bv_b' \mid b$, T_i, f)	Fixates nodes between bisectors or points from the wavefront	$O(K^2 \cdot n \cdot \log(n) + K^3$	Alg. 6

Let us extend set from (11) with new event types: Begin Hole and End Hole process holes; Blind Spike and Inner Wave describe the wavefront reaching the west edge of a hole; Virtual Spike, Outside Spike, Inward Spike and East Corner set the intersection with virtual bisectors. The priority of the first pair equals the abscissa of the edge; the priority of others follows the spike event priority as in (12). The ordering of the events with

the same priority: Blind Spike – Spike – East Edge – Outside Spike – End Hole – Inner Wave – Inward Spike – Virtual Spike – West Edge – East Corner – Begin Hole – Turn. The time complexity of the proposed algorithm is given in Section 4.6.

4.4. Hole Events of Voronoi Diagram. Algorithm 7 defines the Begin Hole event creating the hole shadow from Figure 5. The event has two special cases for intersecting shadows. Let us accept that the algorithm intersects only one shadow layer; then it is always an outer shadow. Then the shadow node with outside virtual bisector is removed from the between set and relocated to the main tree. If this node lies inside another shadow (the second special case), then its shadow node is relocated as well.

Algorithm 8 defines the End Hole event changing the outer shadows. The special case of the End Hole event occurs when the outside bisector belongs to the other active outer shadow. Then the outside bisector of this shadow should also be relocated inside the recreated shadow.

Input $\mathcal{T} \neq \varnothing$, $q \leftarrow Q.pop()$, $\mathcal{L} \leftarrow Priority(q)$, $type(q) \leftarrow$ Begin Hole, $Hole_q$ – hole of the event q
1. $a \leftarrow$ ***Intersect***$(\mathcal{B}v(Hole_q{}^w, Hole_q{}^n), \mathcal{T}, 1)$, $b \leftarrow$ ***Intersect***$(\mathcal{B}v(Hole_q{}^w, Hole_q{}^s), \mathcal{T}, 1)$
2. $\mathcal{T}^{btwn} \leftarrow$ ***Between***$(\mathcal{T}, a, b)$, $\mathcal{T}_{First} \leftarrow \mathcal{T}^{btwn}$.First(), $\mathcal{T}_{Last} \leftarrow \mathcal{T}^{btwn}$.Last()
3. $\mathcal{T}_a{}^{SH} \leftarrow$ ***OnShadow***$(\mathcal{T}_{First}, -1)$, $\mathcal{T}_b{}^{SH} \leftarrow$ ***OnShadow***$(\mathcal{T}_{Last}, 1)$ =
4. **if** $\mathcal{T}_b{}^{SH} \neq \varnothing$ **then** ▷ lower bisector intersect shadow
5. $a^* \leftarrow ep(\mathcal{T}_b{}^{SH})$, $b_{new} \leftarrow$ ***Intersect***$(\mathcal{B}v(Hole_q{}^w, Hole_q{}^s), basis(\mathcal{T}_b{}^{SH}), 1)$
6. $\mathcal{T}^{btwn} \leftarrow \{\mathcal{T}^* \mid \mathcal{T}^* \in$ ***Between***$(basis(\mathcal{T}_b{}^{SH}), a^*, b_{new})\} \cup \mathcal{T}^{btwn}$
7. $\beta v_{In}{}^s \leftarrow$ ***Create***$_{In}(basis(\mathcal{T}_b{}^{SH}), Hole_q{}^w, Hole_q{}^s, b_{new}, -1)$ ▷ lower inward bisector
8. $\beta v_{Out}{}^s \leftarrow$ ***Create***$_{Out}(basis(\mathcal{T}_b{}^{SH}), Hole_q{}^s, b_{new}, -1)$ ▷ lower outside bisector
9. **else** $\beta v_{In}{}^s \leftarrow$ ***Create***$_{In}(\mathcal{T}, Hole_q{}^w, Hole_q{}^s, b, -1)$, $\beta v_{Out}{}^s \leftarrow$ ***Create***$_{Out}(\mathcal{T}, Hole_q{}^s, b, -1)$
10. **if** $\mathcal{T}_a{}^{SH} \neq \varnothing$ **then** ▷ upper bisector intersect shadow
11. $a_{new} \leftarrow$ ***Intersect***$(\mathcal{B}v(Hole_q{}^w, Hole_q{}^n), basis(\mathcal{T}_a{}^{SH}), 1)$, $b^* \leftarrow ep(\mathcal{T}_a{}^{SH})$
12. $\mathcal{T}^{btwn} \leftarrow \mathcal{T}^{btwn} \cup \{\mathcal{T}^* \mid \mathcal{T}^* \in$ ***Between***$(basis(\mathcal{T}_a{}^{SH}), a_{new}, b^*)\}$
13. $\beta v_{In}{}^n \leftarrow$ ***Create***$_{In}(basis(\mathcal{T}_a{}^{SH}), Hole_q{}^w, Hole_q{}^n, a_{new}, 1)$ ▷ upper inward bisector
14 $\beta v_{Out}{}^n \leftarrow$ ***Create***$_{Out}(basis(\mathcal{T}_a{}^{SH}), Hole_q{}^n, a_{new}, 1)$ ▷ upper outside bisector
15. **else** $\beta v_{In}{}^n \leftarrow$ ***Create***$_{In}(\mathcal{T}, Hole_q{}^w, Hole_q{}^n, a, 1)$, $\beta v_{Out}{}^n \leftarrow$ ***Create***$_{Out}(\mathcal{T}, Hole_q{}^n, a, 1)$
16. **for** $\mathcal{T}_{inter}{}^{SH}$ in $\{\mathcal{T}_b{}^{SH} \neq \varnothing, \mathcal{T}_a{}^{SH} \neq \varnothing\}$ **do** ▷ first special case
17. set $\mathcal{T}_{neigh}$ as $prev(\mathcal{T}_{inter}{}^{SH})$ if $\mathcal{T}_{inter}{}^{SH} = \mathcal{T}_b{}^{SH}$ or as $next(\mathcal{T}_{inter}{}^{SH})$ otherwise
18. **if** $end(\mathcal{T}_{inter}{}^{SH}) \neq \mathcal{T}_{neigh}$ **then** ▷ second special case
19. $(\mathcal{T}_{neigh^*}{}^{SH}, \mathcal{T}_{inter}{}^{SH}, \mathcal{T}^{*btwn}) \leftarrow$ ***Swap***$(\mathcal{T}, dupl(\mathcal{T}_{neigh}{}^{SH}), \mathcal{T}_{inter}{}^{SH})$
20. $\mathcal{T}^{btwn} \leftarrow \mathcal{T}^{btwn} \setminus (\mathcal{T}^{*btwn} \cup \{\mathcal{T}_{neigh}{}^{SH}, \mathcal{T}_{inter}{}^{DP}\})$, $\mathcal{T}^{btwn} \leftarrow \{\mathcal{T}_{neigh^*}{}^{SH}\} \cup \mathcal{T}^{btwn}$
21. set $\mathcal{T}_a{}^{SH}$ as $\mathcal{T}_{neigh^*}{}^{SH}$ if $main(\mathcal{T}_{neigh}{}^{SH}) = \mathcal{T}_a{}^{SH}$ ▷ reference the relocated shadow
22. ***Remove***$(basis(\mathcal{T}_{inter}{}^{SH}), \mathcal{T}_{inter}{}^{DP})$, ***Remove***$(\mathcal{T}, \mathcal{T}_{inter}{}^{SH})$
23. $\mathcal{T}^{btwn} \leftarrow \mathcal{T}^{btwn} \setminus \{\mathcal{T}_{inter}{}^{SH}, \mathcal{T}_{inter}{}^{DP}\}$
24. ***Remove***$(basis(\mathcal{T}_{inter}{}^{SH}), \forall \mathcal{T}^* \in \mathcal{T}^{btwn})$
25. ***Insert***$(\mathcal{T}, \beta v_{In}{}^n, \beta v_{In}{}^s)$ ▷ inner shadow with $\mathcal{T}^{btwn}$ inside
26. ***Insert***$(\mathcal{T}, \mathcal{B}v_{Corner}(Hole_q{}^w, Hole_q{}^n))$, ***Insert***$(\mathcal{T}, \mathcal{B}v_{Corner}(Hole_q{}^w, Hole_q{}^s))$ ▷ west corners

27. ***Insert***($\mathcal{T}$, ***Bv*$_{\text{Edge}}$**($Hole_q^n$), $\beta v_{\text{Out}}{}^n$) ▷ upper shadow with north inward and corner inside
28. ***Insert***($\mathcal{T}$, ***Bv*$_{\text{Edge}}$**($Hole_q^s$), $\beta v_{\text{Out}}{}^s$) ▷ lower shadow with south inward and corner inside
29. set $\mathcal{T}_{b^*}{}^{\text{SH}}$ as ***Inject***($\mathcal{T}$, $\mathcal{T}_b{}^{\text{SH}}$, b) if $\mathcal{T}_b{}^{\text{SH}} \neq \varnothing$ ▷ relocate $\mathcal{T}_b{}^{\text{SH}}$ with $\beta v_{\text{Out}}{}^s$ inside
30. set $\mathcal{T}_{a^*}{}^{\text{SH}}$ as ***Inject***($\mathcal{T}$, $\mathcal{T}_a{}^{\text{SH}}$, a) if $\mathcal{T}_a{}^{\text{SH}} \neq \varnothing$ ▷ relocate $\mathcal{T}_a{}^{\text{SH}}$ with $\beta v_{\text{Out}}{}^n$ inside
31. Insert in Q new events for every inserted and relocated node, including the duplicates

Algorithm 7. Begin Hole event processing

Input $\mathcal{T} \neq \varnothing$, $q \leftarrow Q.pop()$, $\mathcal{L} \leftarrow Priority(q)$, $type(q) \leftarrow$ End Hole, $Hole_q$ – hole of the event q

1. set $\mathcal{T}_{\text{NEdge}}{}^{\text{SH}}$ as node of ***Bv*$_{\text{Edge}}$**($Hole_q^n$), set $\mathcal{T}_{\text{SEdge}}{}^{\text{SH}}$ as node of ***Bv*$_{\text{Edge}}$**($Hole_q^s$)
2. **for** $\mathcal{T}_{\text{Edge}}{}^{\text{SH}}$ in {$\mathcal{T}_{\text{NEdge}}{}^{\text{SH}}$, $\mathcal{T}_{\text{SEdge}}{}^{\text{SH}}$} **do** ▷ for outer shadows
3. $\mathcal{T}_{\text{Out}}{}^{\text{SH}} \leftarrow end(\mathcal{T}_{\text{Edge}}{}^{\text{SH}})$, ***Remove***($dupl(\mathcal{T}_{\text{Edge}}{}^{\text{SH}})$, $dupl(\mathcal{T}_{\text{Edge}}{}^{\text{SH}})$), ***Remove***($\mathcal{T}$, $\mathcal{T}_{\text{Edge}}{}^{\text{SH}}$)
4. **if** $\mathcal{T}_{\text{Edge}}{}^{\text{SH}} = \mathcal{T}_{\text{NEdge}}{}^{\text{SH}}$ **then**
5. $\mathcal{T}_{\text{neigh}} \leftarrow next(\mathcal{T}_{\text{Edge}}{}^{\text{SH}})$, $Hole_q{}^u \leftarrow Hole_q{}^n$, $f \leftarrow 1$
6. **else** $\mathcal{T}_{\text{neigh}} \leftarrow prev(\mathcal{T}_{\text{Edge}}{}^{\text{SH}})$, $Hole_q{}^u \leftarrow Hole_q{}^s$, $f \leftarrow -1$
7. $\mathcal{T}_{\text{Corner}}{}^{\text{SH}} \leftarrow$ ***Insert*$_{\text{SH}}$**($\mathcal{T}$, ***Bv*$_{\text{Corner}}$**($Hole_q{}^e$, $Hole_q{}^u$)), $end(\mathcal{T}_{\text{Corner}}{}^{\text{SH}}) \leftarrow \mathcal{T}_{\text{Out}}{}^{\text{SH}}$
8. **if** $\mathcal{T}_{\text{neigh}} \neq \mathcal{T}_{\text{Out}}{}^{\text{SH}}$ **then** ▷ the special case
9. ($\mathcal{T}_{\text{neigh}^*}{}^{\text{SH}}$, $\mathcal{T}_{\text{Out}}{}^{\text{SH}}$, $\mathcal{T}^{\text{btwn}}$) $\leftarrow$ ***Swap***($\mathcal{T}$, $dupl(\mathcal{T}_{\text{neigh}}{}^{\text{SH}})$, $\mathcal{T}_{\text{Out}}{}^{\text{SH}}$)
10. $\beta v_{\text{In}} \leftarrow$ ***Create*$_{\text{In}}$**($\mathcal{T}$, $Hole_q{}^e$, $ep(\mathcal{T}_{\text{Out}}{}^{\text{SH}})$, f)
11. ***Remove***($basis(\mathcal{T}_{\text{Out}}{}^{\text{SH}})$, $dupl(\mathcal{T}_{\text{Out}}{}^{\text{SH}})$), ***Remove***($\mathcal{T}$, $\mathcal{T}_{\text{Out}}{}^{\text{SH}}$)
12. $\mathcal{T}_{\text{In}}{}^{\text{SH}} \leftarrow$ ***Insert*$_{\text{SH}}$**($\mathcal{T}$, βv_{In}), $end(\mathcal{T}_{\text{In}}{}^{\text{SH}}) \leftarrow \mathcal{T}_{\text{Corner}}{}^{\text{SH}}$
13. Insert in Q new events for every inserted and relocated node, including the duplicates

Algorithm 8. End Hole event processing

Figure 8 illustrates the first and second special case of Begin Hole event and the special case of End Hole event.

Algorithm 9 defines the Blind Spike event handling intersection between a real node and a blind node at a point p_q. If both nodes are blind, then the algorithm only removes them; else it creates a new blind node in the same or in another position along the edge.

Algorithm 10 defines the Inner Wave event of the wavefront reaching the west edge of the hole. It occurs if some node with a real bisector bv_q reaches the west edge of $Hole_q$; then the wavefront segment with the bisector bv_q should be fixated partly or entirely on the west edge at the current moment of the sweep line. To track the Inner Wave event during the insertion of the new events for created flat nodes or relocated flat nodes from the between set in the Begin Hole event, we should check if they lie in the inner shadow using *Front* and if they have an intersection with the west edge of the hole. If p_q is the point of intersection, T_q is any owner of bv_q, then the priority of the Inner Wave event is calculated by equation (12): $x_q + \mathrm{d}(p_q, T_q)$.

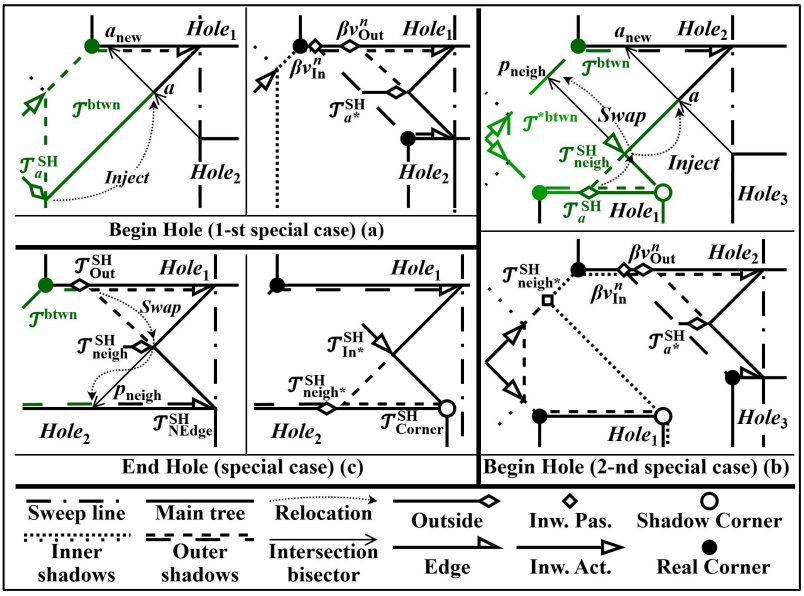


Fig. 8. Special cases of hole events: first a) and second, b) special cases of Begin Hole event, c) special case of End Hole; changed nodes shown in green

Input $q \leftarrow Q.pop()$, $\mathcal{L} \leftarrow Priority(q)$, $type(q) \leftarrow$ Blind Spike, $\mathcal{T}_{q1}$, $\mathcal{T}_{q2}$: $\mathcal{T}_{q2} \leftarrow next(\mathcal{T}_{q1})$, p_q
1. **if** $type(\mathcal{T}_{q1})$ = BL **and** $type(\mathcal{T}_{q2})$ = BL **then** ▷ both bisectors blind
2. ***Remove***$(basis(\mathcal{T}_{q1}^{BL}), \mathcal{T}_{q2})$, ***Remove***$(basis(\mathcal{T}_{q1}^{BL}), \mathcal{T}_{q1})$, **End** ▷ only remove them
3. **if** $type(\mathcal{T}_{q1})$ = BL **then**
4. $\mathcal{T}^{BL} \leftarrow \mathcal{T}_{q1}^{BL}$, $\mathcal{T}' \leftarrow \mathcal{T}_{q2}$, $\mathcal{T}_{neigh} \leftarrow next(\mathcal{T}')$, $p_{neigh} \leftarrow ep(\mathcal{T}_{neigh})$, $Hole^w \in owners(\mathcal{T}_{q1})$
5. **else** $\mathcal{T}^{BL} \leftarrow \mathcal{T}_{q2}^{BL}$, $\mathcal{T}' \leftarrow \mathcal{T}_{q1}$, $\mathcal{T}_{neigh} \leftarrow prev(\mathcal{T}')$, $p_{neigh} \leftarrow ep(\mathcal{T}_{neigh})$, $Hole^w \in owners(\mathcal{T}_{q2})$
6. **while** $\mathcal{T}_{neigh} \neq \varnothing$ **and** $x_{neigh} = x_q$ **do** ▷ while nodes with same abscissa
7. $T_i^{\mu} \leftarrow owners(\mathcal{T}') \cap owners(\mathcal{T}_{neigh})$, ***Voronoi***$(T_i, (ep(\mathcal{T}'), p_{neigh}))$
8. ***Remove***$(basis(\mathcal{T}^{BL}), \mathcal{T}')$, $\mathcal{T}' \leftarrow \mathcal{T}_{neigh}$
9. set $\mathcal{T}_{neigh}$ as $next(\mathcal{T}')$ if $type(\mathcal{T}_{q1})$ = BL or as $prev(\mathcal{T}')$ otherwise
10. **if** $type(\mathcal{T}') \notin$ {BL, DP} **then**
11. $T_i^{\mu} \leftarrow owners(\mathcal{T}') \cap owners(\mathcal{T}_{neigh})$, ***Remove***$(basis(\mathcal{T}^{BL}), \mathcal{T}')$
12. ***Remove***$(basis(end(\mathcal{T}^{BL})), \mathcal{T}^{BL})$, $\mathcal{T}_{new}^{BL} \leftarrow$ ***Insert***$_{BL}(basis(end(\mathcal{T}^{BL})),$ ***Bv***$(T_i^{\mu}, Hole^w))$
13. $end(\mathcal{T}_{new}^{BL}) \leftarrow end(\mathcal{T}^{BL})$
14. Insert in Q new spike events for $\mathcal{T}_{new}^{BL}$

Algorithm 9. Blind Spike event processing

Input $q \leftarrow Q.pop()$, $\mathcal{L} \leftarrow Priority(q)$, $type(q) \leftarrow$ Inner Wave, $\mathcal{T}_q$: $bisector(\mathcal{T}_q) \leftarrow bv_q$, $Hole_q^w$
1. check if $\mathcal{T}_q$ is removed then ignore this event
2. $\mathcal{T}' \leftarrow prev(\mathcal{T}_q)$, $p' \leftarrow ep(\mathcal{T}')$, $\mathcal{T}^{Wave} \leftarrow \{\mathcal{T}_q\}$, set x_q as abscissa of hole's west edge
3. add to $\mathcal{T}^{Wave}$ all nodes from left to right with current abscissa = x_q by *left* and *right* from $\mathcal{T}_q$

4. remove first node from $\mathcal{T}^{\text{Wave}}$ if it's duplicate or blind $\mathcal{T}^{\text{Wave}}$.Remove($\mathcal{T}^{\text{Wave}}$.First())
5. remove last node from $\mathcal{T}^{\text{Wave}}$ if it's duplicate or blind $\mathcal{T}^{\text{Wave}}$.Remove($\mathcal{T}^{\text{Wave}}$.Last())
6. $\mathcal{T}_{\text{Low}} \leftarrow \mathcal{T}^{\text{Wave}}$.First(), $\mathcal{T}_{\text{Up}} \leftarrow \mathcal{T}^{\text{Wave}}$.Last(), $\mathcal{T}_{\text{Front}} \leftarrow \mathit{Front}(\mathcal{T}_q)$
7. $T_{\text{Low}}{}^u \leftarrow \mathit{owners}(\mathcal{T}_{\text{Low}}) \cap \mathit{owners}(\mathit{next}(\mathcal{T}_{\text{Low}}))$, $T_{\text{Up}}{}^u \leftarrow \mathit{owners}(T_{\text{Up}}) \cap \mathit{owners}(\mathit{prev}(T_{\text{Up}}))$
8. **for** $\mathcal{T}^* \in \mathcal{T}^{\text{Wave}} \setminus \mathcal{T}^{\text{Wave}}$.Last() **do** ▷ fixate segments lying on edge of hole
9. $T_i{}^u \leftarrow \mathit{owners}(\mathcal{T}^*) \cap \mathit{owners}(\mathit{next}(\mathcal{T}^*))$, ***Voronoi***$(T_i, (\mathit{ep}(\mathcal{T}^*), \mathit{ep}(\mathit{next}(\mathcal{T}^*))))$
10. $\mathcal{T}_{\text{UpNew}} \leftarrow$ ***Insert***$_{\text{F}}(\mathcal{T}_{\text{Front}}, \mathit{Bv}(T_{\text{Up}}{}^u, \mathit{Hole}_q{}^w))$, $\mathcal{T}_{\text{LowNew}} \leftarrow$ ***Insert***$_{\text{F}}(\mathcal{T}_{\text{Front}}, \mathit{Bv}(T_{\text{Low}}{}^u, \mathit{Hole}_q{}^w))$
11. ***Remove***$(\mathcal{T}_{\text{Front}}, \forall \mathcal{T}^* \in \mathcal{T}^{\text{Wave}})$
12. Insert in $\mathcal{Q}$ new events for new flat nodes on hole west edge $\mathcal{T}_{\text{UpNew}}$ and $\mathcal{T}_{\text{LowNew}}$

Algorithm 10. Inner Wave event processing.

4.5. Virtual Spike Events of the Voronoi Diagram. The Virtual Spike event occurs when a bisector intersects a virtual bisector. If q_{VSp} is a Virtual Spike event, $\mathscr{b}v_1$ and $\mathscr{b}v_2$ are intersecting bisectors, $p_{\text{VSp}} = \mathscr{b}v_1 \cap \mathscr{b}v_2$ is the intersection point, then the priority of the Virtual Spike event is:

$$Priority(q_{\text{VSp}}) = x_{\text{VSp}} + d(p_{\text{VSp}}, R_{\text{VSp}}),$$

where R_{VSp} is a common support edge owner if the bisectors have such, or the only support edge owner if any bisector has such, or a hole edge owner which induces an inward or outside virtual bisector.

Figure 9 illustrates examples of every virtual intersection event types.

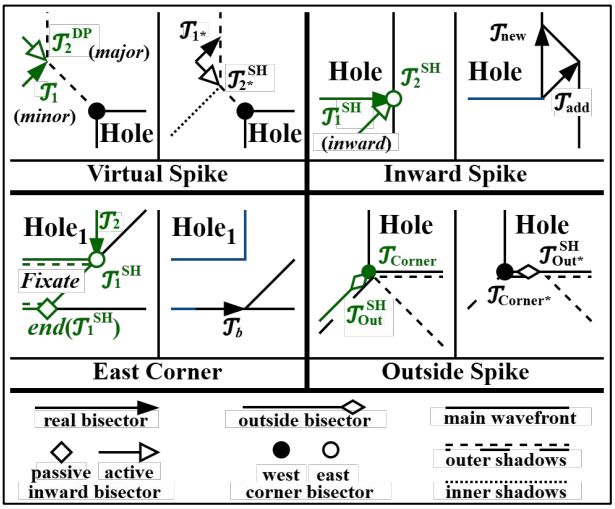


Fig. 9. Examples of intersection events with virtual bisectors

The intersection events are specified at the insertion moment. Figure 10 describes a specification of intersection events. The type of a virtual corner bisector can be defined by abscissa equality with the corresponding edge of the hole. Label "X" is shown for the pairs that should never have an intersection event.

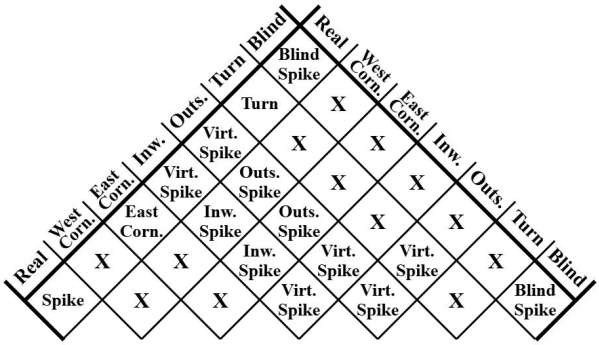


Fig. 10. Specification of intersection events between different types of bisectors

Algorithm 11 defines the Virtual Spike event. If the first bisector represents a duplicate node, then the second bisector exits the shadow, relocating both of them. Else one enters the shadow of the other. Both bisectors must not be removed or the event should be ignored.

Algorithm 12 defines the first special case with intersection between outside and corner virtual bisectors. If the outside bisector moves along the east edge, then the algorithm relocates it in the inward direction from the vertex including the corner and its paired inward bisector to the shadow. If the outside bisector reaches the east corner by the horizontal edge, then it exits the shadow, includes the corner and its paired shadow node into the shadow and relocates this corner pair in the inward direction. Else the outside bisector includes the corner node into the shadow and redirects itself along the edge. Let us call it an Outside Spike event.

Input $q = \mathcal{Q}.pop()$, $\mathcal{L} = Priority(q)$, $type(q)$ = Virtual Spike, $\mathcal{T}_1$, $\mathcal{T}_2$: $\mathcal{T}_2 = next(\mathcal{T}_1)$, p_{VSp}
1. check if $\mathcal{T}_1$ or $\mathcal{T}_2$ are removed then ignore this event
2. set ($\mathcal{T}_{major}$, $\mathcal{T}_{minor}$) as ($\mathcal{T}_2$, $\mathcal{T}_1$) if ($type(\mathcal{T}_2)$ = DP or $type(\mathcal{T}_1)$ = F) or vice versa otherwise
3. set all $\mathcal{T}_{Front}$, $\mathcal{T}_{from}$, $\mathcal{T}_{into}$ as ***Front***$(\mathcal{T}_{major})$, set all p_{minor^*}, p_{major^*} as p_{VSp}, $t \leftarrow type(\mathcal{T}_{minor})$
4. **if** $type(\mathcal{T}_{major})$ = DP **then** ▷ $\mathcal{T}_{minor}$ exits shadow $\mathcal{T}_{major}$
5. $\mathcal{T}_{major}^{SH} \leftarrow main(\mathcal{T}_{major}^{DP})$, $\mathcal{T}_{from} \leftarrow basis(\mathcal{T}_{major}^{SH})$, $p_{minor^*} \leftarrow p_{VSp}$, $p_{major^*} \leftarrow p_{VSp}$
6. **if** $\mathcal{T}_{major} = \mathcal{T}_2$ **then** ▷ points for $\mathcal{T}_{major}$ relocation
7. $p_{inside} \leftarrow ep(prev(\mathcal{T}_{major}^{SH}))$, $p_{neigh} \leftarrow ep(prev(\mathcal{T}_{minor}))$
8. **else** $p_{inside} \leftarrow ep(next(\mathcal{T}_{major}^{SH}))$, $p_{neigh} \leftarrow ep(next(\mathcal{T}_{minor}))$

9. set p_{major^*} as p_{inside} if $p_{inside} \in \mathcal{B}v_{insd}(\mathcal{T}_{major}^{SH})$
10. set p_{major^*} as p_{neigh} if ($p_{neigh} \in \mathcal{B}v_{insd}(\mathcal{T}_{major}^{SH})$ and $x_{neigh} < x_{major^*}$)
11. **if** t = SH **then** ▷ need to relocate $\mathcal{T}_{minor}$
12. **if** $\mathcal{T}_{major} = \mathcal{T}_2$ **then**
13. $p_{inside} \leftarrow ep(next(\mathcal{T}_{major}^{SH}))$, $p_{neigh} \leftarrow ep(next(dupl(\mathcal{T}_{minor}^{SH}))$
14. **else** $p_{inside} \leftarrow ep(prev(\mathcal{T}_{major}^{SH}))$, $p_{neigh} \leftarrow ep(prev(dupl(\mathcal{T}_{minor}^{SH}))$
15. set p_{minor^*} as p_{inside} if $p_{inside} \in \mathcal{B}v_{insd}(\mathcal{T}_{minor}^{SH})$
16. set p_{minor^*} as p_{neigh} if ($p_{neigh} \in \mathcal{B}v_{insd}(\mathcal{T}_{minor}^{SH})$ and $x_{neigh} < x_{minor^*}$)
17. **else** $\mathcal{T}_{into} \leftarrow basis(\mathcal{T}_{major}^{SH})$, $\mathcal{B}v_{major^*} \leftarrow$ ***Relocate***$(\mathcal{T}_{Front}, \mathcal{T}_{major}^{SH}, p_{VSp})$
18. **if** t = SH **then** ▷ find relocated $\mathcal{T}_{minor}$
19. $\mathcal{B}v_{minor^*} \leftarrow$ ***Relocate***$(\mathcal{T}_{into}, \mathcal{T}_{minor}^{SH}, p_{minor^*})$, ***Remove***$(basis(\mathcal{T}_{minor}^{SH}), dupl(\mathcal{T}_{minor}))$
20. **else** $\mathcal{B}v_{minor^*} \leftarrow$ ***Bv***$(owners(\mathcal{T}_{minor}))$
21. ***Remove***$(\mathcal{T}_{from}, \mathcal{T}_{minor})$, ***Remove***$(basis(\mathcal{T}_{major}^{SH}), \mathcal{T}_{major}^{DP})$, ***Remove***$(\mathcal{T}_{Front}, \mathcal{T}_{major}^{SH})$
22. $\mathcal{T}_{minor^*} \leftarrow$ ***Insert***$_t(\mathcal{T}_{into}, \mathcal{B}v_{minor^*})$
23. set $\mathcal{B}v_{major^*}$ as ***Relocate***$(\mathcal{T}_{Front}, \mathcal{T}_{major}^{SH}, p_{major^*})$ if $type(\mathcal{T}_{major})$ = DP
24. $\mathcal{T}_{major^*}^{SH} \leftarrow$ ***Insert***$_{SH}(\mathcal{T}_{from}, \mathcal{B}v_{major^*})$
25. Insert in $\mathcal{Q}$ new events for $\mathcal{T}_{major^*}^{SH}$ and its duplicate, $\mathcal{T}_{minor^*}$ and its duplicate if isn't flat

Algorithm 11. Virtual Spike event processing

Input $q = \mathcal{Q}.pop()$, $\mathcal{L} = Priority(q)$, $type(q)$ = Outside Spike, $\mathcal{T}_1$, $\mathcal{T}_2$: $\mathcal{T}_2 = next(\mathcal{T}_1)$, p_{VSp}
1. check if $\mathcal{T}_1$ or $\mathcal{T}_2$ are removed then ignore this event
2. **if** $type(\beta v_1)$ = C **then**
3. $\mathcal{T}_{Out}^{SH} \leftarrow \mathcal{T}_2^{SH}$, $\mathcal{T}_{Corner} \leftarrow \mathcal{T}_1$, $\beta v_{Out} \leftarrow bisector(\mathcal{T}_2)$, $\beta v_{Corner} \leftarrow bisector(\mathcal{T}_1)$, $f \leftarrow -1$
4. **else** $\mathcal{T}_{Out}^{SH} \leftarrow \mathcal{T}_1^{SH}$, $\mathcal{T}_{Corner} \leftarrow \mathcal{T}_2$, $\beta v_{Out} \leftarrow bisector(\mathcal{T}_1)$, $\beta v_{Corner} = bisector(\mathcal{T}_2)$, $f \leftarrow 1$
5. $Hole_{k1}^{u1} \in owners(\beta v_{Out})$: $Hole_{k1}^{u1} \in Hole_{k1}$ ▷ hole edge inducing outside bisector
6. **if** $type(\mathcal{T}_{Corner})$ = DP **then** ▷ east corner by horizontal edge case
7. $\mathcal{T}_{Corner}^{SH} \leftarrow main(\mathcal{T}_{Corner}^{DP})$, $\mathcal{T}_{End}^{SH} \leftarrow end(\mathcal{T}_{Corner}^{SH})$, $\mathcal{T}_{Front} \leftarrow$ ***Front***$(\mathcal{T}_1)$
8. $(\mathcal{T}_{End^*}^{SH}, \mathcal{T}_{Out}^{SH}, \mathcal{T}^{btwn}) \leftarrow$ ***Swap***$(\mathcal{T}_{Front}, dupl(\mathcal{T}_{End}^{SH}), \mathcal{T}_{Out}^{SH})$
9. $\beta v_{Out^*} \leftarrow$ ***Create***$_{Out}(\mathcal{T}_{Front}, Hole_{k1}^{u1}, ep(\mathcal{T}_{Corner}), f)$
10. ***Remove***$(basis(\mathcal{T}_{Corner}^{SH}), dupl(\mathcal{T}_{Corner}^{SH}))$, ***Remove***$(\mathcal{T}_{Front}, \mathcal{T}_{Corner}^{SH})$
11. $\mathcal{T}_{Corner^*}^{SH} \leftarrow$ ***Insert***$_{SH}(end(\mathcal{T}_{Out}^{SH}), \beta v_{Corner})$
12. ***Remove***$(basis(\mathcal{T}_{Out}^{SH}), dupl(\mathcal{T}_{Out}^{SH}))$, ***Remove***$(\mathcal{T}_{Front}, \mathcal{T}_{Out}^{SH})$
13. $\mathcal{T}_{Out^*}^{SH} \leftarrow$ ***Insert***$_{SH}(\mathcal{T}_{Front}, \beta v_{Out^*})$
14. **else if** $type(\mathcal{T}_{Corner}) \neq$ SH **then** ▷ west corner case
15. $\mathcal{T}_{Front} \leftarrow$ ***Front***$(\mathcal{T}_1)$, $\beta v_{Out^*} =$ ***Create***$_{Out}(\mathcal{T}_{Front}, Hole_{k1}^{u1}, ep(\mathcal{T}_{Corner}), f)$
16. ***Remove***$(\mathcal{T}_{Front}, \mathcal{T}_{Corner})$, $\mathcal{T}_{Corner^*} =$ ***Insert***$_F(end(\mathcal{T}_{Out}^{SH}), \beta v_{Corner})$
17. ***Remove***$(basis(\mathcal{T}_{Out}^{SH}), dupl(\mathcal{T}_{Out}^{SH}))$, ***Remove***$(\mathcal{T}_{Front}, \mathcal{T}_{Out}^{SH})$
18. $\mathcal{T}_{Out^*}^{SH} \leftarrow Insert_{SH}(\mathcal{T}_{Front}, \beta v_{Out^*})$
19. ▷ east corner by east edge case
20. **else** $\mathcal{T}_{End}^{SH} \leftarrow end(\mathcal{T}_{Corner}^{SH})$, $\mathcal{T}_{End}^{DP} \leftarrow dupl(\mathcal{T}_{End}^{SH})$, $\mathcal{T}_{Front} \leftarrow$ ***Front***$(\mathcal{T}_1)$
21. ***Remove***$(basis(\mathcal{T}_{Out}^{SH}), dupl(\mathcal{T}_{Out}^{SH}))$, ***Remove***$(\mathcal{T}_{Front}, \mathcal{T}_{Out}^{SH})$
22. set $\mathcal{T}_{neigh}$ as $next(\mathcal{T}_{End}^{DP})$ if $type(\beta v_1)$ = C or as $prev(\mathcal{T}_{End}^{DP})$ otherwise
23. ***Remove***$(basis(\mathcal{T}_{Corner}^{SH}), dupl(\mathcal{T}_{Corner}^{SH}))$, ***Remove***$(\mathcal{T}_{Front}, \mathcal{T}_{Corner}^{SH})$
24. $\mathcal{T}_{Corner^*}^{SH} \leftarrow$ ***Insert***$_{SH}(end(\mathcal{T}_{Out}^{SH}), \beta v_{Corner})$
25. **if** $type(\mathcal{T}_{neigh})$ = SH **then** ▷ if intersects another shadow

26. $(\mathcal{T}_{\text{End}}^{\text{SH}}, \mathcal{T}_{\text{neigh*}}^{\text{SH}}, \mathcal{T}^{\text{btwn}}) =$ ***Swap***$(\mathcal{T}, \mathcal{T}_{\text{End}}^{\text{DP}}, \mathcal{T}_{\text{neigh}}^{\text{SH}})$, $\mathcal{T}_{\text{Front}} = basis(\mathcal{T}_{\text{neigh*}}^{\text{SH}})$
27. $\mathcal{T}_{\text{Out*}}^{\text{SH}} =$ ***Inject***$(\mathcal{T}_{\text{Front}}, \mathcal{T}_{\text{Out}}^{\text{SH}}, ep(\mathcal{T}_{\text{End}}^{\text{SH}}))$
28. Insert in $\mathcal{Q}$ new events for all created and relocated nodes, including duplicates;

Algorithm 12. Outside Spike event processing

Algorithm 13 defines the second special case of inward and corner virtual bisectors. For the east corner, the algorithm removes the shadow and creates new real bisectors from the east vertex of the hole. Else the algorithm creates a real bisector along the horizontal edge and a blind node along the west edge. Both happen only if the corner virtual bisector does not lie on the slab edge. Let us call this special case an Inward Spike event.

Input $q = \mathcal{Q}.pop()$, $\mathcal{L} = Priority(q)$, $type(q) =$ Inward Spike, $\mathcal{T}_1, \mathcal{T}_2$: $\mathcal{T}_2 = next(\mathcal{T}_1)$, p_{VSp}
1. check if $\mathcal{T}_1$ or $\mathcal{T}_2$ are removed then ignore this event
2. **if** $type(bisector(\mathcal{T}_1)) =$ C **then**
3. $\quad \mathcal{T}_{\text{inw}}^{\text{SH}} \leftarrow \mathcal{T}_2^{\text{SH}}$, $\mathcal{T}_{\text{Corner}} \leftarrow \mathcal{T}_1$, $p_{\text{neigh}} \leftarrow ep(next(\mathcal{T}_2))$
4. **else** $\mathcal{T}_{\text{inw}}^{\text{SH}} \leftarrow \mathcal{T}_1^{\text{SH}}$, $\mathcal{T}_{\text{Corner}} \leftarrow \mathcal{T}_2$, $p_{\text{neigh}} \leftarrow ep(prev(\mathcal{T}_1))$
5. check if corner virtual bisector lies on the slab edge then ignore this event
6. $(Hole^{u1}, Hole^{u2}, T^{u3}) \leftarrow owners(\mathcal{T}_{\text{inw}}^{\text{SH}})$, $\mathcal{T}_{\text{Front}} \leftarrow$ ***Front***$(\mathcal{T}_1)$, ***Remove***$(\mathcal{T}_{\text{inw}}^{\text{SH}}, dupl(\mathcal{T}_{\text{inw}}^{\text{SH}}))$
7. set $Hole^{\text{ed}}$ as $Hole^{u2}$ if ***Form***$(\mathcal{T}_{\text{inw}}^{\text{SH}}) =$ Inner or as $Hole^{u1}$ otherwise
8. **if** $Form(\mathcal{T}_{\text{inw}}^{\text{SH}}) =$ Inner **then**
9. $\quad \mathcal{T}_w \leftarrow$ ***Insert***$_{\text{BL}}(end(\mathcal{T}_{\text{inw}}^{\text{SH}})$, ***Bv***$(T^{u3}, Hole^{u1}))$, $end(\mathcal{T}_w^{\text{BL}}) \leftarrow end(\mathcal{T}_{\text{inw}}^{\text{SH}})$
10. **else** remove flat node on the edge and both duplicates from $basis(\mathcal{T}_{\text{inw}}^{\text{SH}})$
11. ***Remove***$(\mathcal{T}_{\text{Front}}, \mathcal{T}_1)$, ***Remove***$(\mathcal{T}_{\text{Front}}, \mathcal{T}_2)$, $\mathcal{T}_{\text{new}} \leftarrow$ ***Insert***$_{\text{F}}(\mathcal{T}$, ***Bv***$(T^{u3}, Hole^{\text{ed}}))$
12. **if** (***Form***$(\mathcal{T}_{\text{inw}}^{\text{SH}}) =$ Inner **and** $x_{\text{neigh}} < x_{\text{VSp}}$) **or** (***Form***$(\mathcal{T}_{\text{inw}}^{\text{SH}}) =$ Outer **and** $x_{\text{neigh}} \le x_{\text{VSp}}$) **then**
13. $\quad \mathcal{T}_{\text{add}} \leftarrow$ ***Insert***$_{\text{F}}(\mathcal{T}_{\text{Front}}, Bv(T^{u3}, p_{\text{VSp}}))$ ▷ additional bisector as in (13)
14. Insert in $\mathcal{Q}$ new events for all created flat and blind nodes

Algorithm 13. Inward Spike event processing

Algorithm 14 defines the third special case with intersection of a real bisector along the hole's east edge and its corner. The algorithm fixates the shadow and creates a new flat node using the intersection of the bisector in the inward direction from the corner. Let us call this special case an East Corner event.

Input $q = \mathcal{Q}.pop()$, $\mathcal{L} = Priority(q)$, $type(q) =$ East Corner, $\mathcal{T}_1, \mathcal{T}_2$: $\mathcal{T}_2 = next(\mathcal{T}_1)$, p_{VSp}
1. check if $\mathcal{T}_1$ or $\mathcal{T}_2$ are removed then ignore this event
2. $\mathcal{T}_{\text{Front}} \leftarrow$ ***Front***$(\mathcal{T}_1)$, set $T_{\text{fix}} \in T$ such that $T_{\text{fix}}^u \in owners(\mathcal{T}_1) \cup owners(\mathcal{T}_2)$
3. **if** $type(\mathcal{T}_1) =$ SH **then**
4. $\quad$ ***Fixate***$(\mathcal{T}_{\text{Front}}, ep(\mathcal{T}_1),$ ***Bv***$_{\text{insd}}(end(\mathcal{T}_1^{\text{SH}})), T_{\text{fix}}, 1)$ ▷ south east corner case
5. **else** ***Fixate***$(\mathcal{T}_{\text{Front}},$ ***Bv***$_{\text{insd}}(end(\mathcal{T}_2^{\text{SH}})), ep(\mathcal{T}_2), T_{\text{fix}}, 1)$ ▷ north east corner case

Algorithm 14. East Corner event processing

4.6. Time complexity of the events and proposed algorithm. Each support creates 4 nodes and each hole creates 14 nodes. Therefore the total count of nodes is $n = 4 \cdot N + 14 \cdot K$; thus, the time complexity of the proposed algorithm $O(n)$ equals $O(N + K)$. Table 4 shows the time complexity of a single execution for each event type and the estimation for the count of events of each type happening in the proposed algorithm.

Table 4. Time complexity and number of events execution

Event	Purpose	Key operation	Single exec.	Count
Initial	Initialize the wavefront	Insert border, turn bisectors; initiate supports, holes events	$O((N+K)\cdot \log(N+K))$	$=1$
West Edge	Process a new support	Fixate nodes in wavefront; insert new real bisectors	$O(K^2 \cdot N \cdot \log(N+K) + K^3 \cdot \log(N+K)$	$=N$
East Corner	Intersection of a real bisector and hole's east vertex	Fixate nodes in wavefront; insert new real bisectors		$\leq 2 \cdot K$
Blind Spike	Intersection with a blind node	Remove nodes; recreate blind node along hole edge	$O((N+K)\cdot \log(N+K))$	$\leq 2 \cdot N + 2 \cdot K$
Begin Hole	Process a new hole	Create virtual bisectors; enclose wavefront nodes in new shadows; relocate intersected shadow nodes	$O((N+K)\cdot \log(N+K))$	$=K$
End Hole	End hole's processing	Transform shadow nodes of hole's outer shadows	$O((N+K)\cdot \log(N+K))$	$=K$
Outside Spike	Intersection of an outside bisector and hole's vertex	Relocate outside bisector, related shadow node of vertex	$(N+K)\cdot \log(N+K)$	$\leq 2 \cdot K$
Inner Wave	Process wavefront reaching hole's west edge	Fixate nodes reaching west edge; create new bisectors along the edge	$(N+K)\cdot \log(N+K)$	$<2 \cdot N \cdot K$
East Edge	End support's processing	Remove bisectors related to edges; create new from east vertices	$\log(N+K)$	$=N$
Spike	Intersection of real bisectors	Remove bisectors; create a new one	$\log(N+K)$	$\leq 2 \cdot N + 2 \cdot K$
Turn	Intersection of a real and a turn bisectors	Remove bisectors; create a new real bisector by slab's horizontal edge	$\log(N+K)$	≤ 2
Virtual Spike	Intersection with a virtual bisector	Remove bisectors; recreate both bisectors on needed shadow levels	$\log(N+K)$	$\leq 2 \cdot N + 2 \cdot K$
Inward Spike	Intersection of an inward bisector and its hole's vertex	Remove bisectors; create a real and an additional bisectors from vertex	$\log(N+K)$	$\leq 4 \cdot K$

Then the worst time complexity of the complete proposed algorithm is the time complexity of the fixation events $O((K^2 \cdot N^2 + K^3 \cdot N + K^4) \cdot \log(N+K))$, where N is the count of supports, K is the count of holes.

The proposed events cover most of the holes processing cases. The proposed shadow structure maintains holes in the EP&DTL algorithm without violating its integrity. The time consumption of the algorithm highly depends on the shadow hierarchy in the diagram and its concatenation and mutual relations. Therefore, the time complexity, which determines the practical applicability of the proposed algorithm will be analyzed in the computational experiment.

5. Computational Experiments. This section examines the time consumption of the proposed algorithm. Additionally, the last subsection contains a preliminary analysis of the correlation estimation between some heuristic parameters of the obtained Voronoi cells and evaluated deformation values and a discussion of the current limitations.

5.1. Time consumption experiment. The first experiment tests generated plans, grouped by the count of supports, for time consumption. Every group consists of 1000 generated plans. Table 5 describes the settings of the plans generator. The dimensions of supports and holes are selected to have a linear dependency on the count of supports and the slab area. The difference between maximum dimensions of holes correlates with the ratio of the potential maximum sum of areas of holes to the slab area. Algorithm 15 describes the generating process following the geometric assumptions.

Support dimensions depend on their direction, while the width is constant. Tests use C# .NET Framework 4.8 in Release mode and run on an Intel(R) Core(TM) i7-12650H @ 2.30GHz with 16 GB of RAM.

The experiment compares execution time for generated plans with different count of holes: 0%, 8%, 20%, 32%, 44% and 56% of the number of supports. The proposed algorithm involves the EP&DTL algorithm from [20] as a reimplementation in C#. Additionally, the experiment compares time consumption with the previous algorithm used for deformation comparative analysis, namely, span determination algorithm [12].

Table 5. Setting parameters for generating support-hole plans (in m)

Count of supports N	Slab dimensions $L\times W$	Dimensions of supports		Dimensions of holes					
		Width mT	Max length MT	Min mH	Max MH by sets with K as % of N				
					8%	20%	32%	44%	56%
25	20×20	0.25	5	1	9	5	3.5	3	2.25
50	30×30	0.25	5.75	1	9.5	5.5	4	3.5	2.75
75	40×40	0.25	6.5	1	10	6	4.5	4	3.25
100	50×50	0.25	7.25	1	10.5	6.5	5	4.5	3.75
125	60×60	0.25	8	1	11	7	5.5	5	4.25
150	70×70	0.25	8.75	1	11.5	7.5	6	5.5	4.75
175	80×80	0.25	9.5	1	12	8	6.5	6	5.25
200	90×90	0.25	10.25	1	12.5	8.5	7	6.5	5.75

Input *Slab*, *N*, *K*, (*MT*, *mT*) – support dimensions, (*MH*, *mH*) – hole dimensions, $Rect \leftarrow \{\varnothing\}$

1. **while** $|Rect| < N$ **do** ▷ supports generating
2. set $d_{new} \in [mT, MT]$ by random with step mT, set $f_{new} \in \{0, 1\}$ by random
3. set (l_{new}, w_{new}) as (d_{new}, mT) if $f_{new} = 0$ or (l_{new}, w_{new}) as (mT, d_{new}) otherwise
4. set $x_{new} \in [0, L - l_{new}]$ and $y_{new} \in [0, W - w_{new}]$ by random with step TW
5. $T_{new} \leftarrow <(x_{new}, y_{new}), (l_{new}, w_{new})>$, $T^* \leftarrow \{T_{new}\}$ ▷ new generated rectangle
6. check if any support from *Rect* contains T_{new} or is contained by T_{new} then **Continue**
7. **while** $\exists T_i \in Rect, \exists T' \in T^*: T_i \cap T' \neq \varnothing$ **and** $(\nexists T^u = T_i \cap T': T^u \in T_i, T^u \in T')$ **do**

split T_i by T' into set T_i^*, remove T_i from *Rect*, add rectangles from T_i^* to *Rect*
split T' by T_i without intersection $T_i \cap T'$ into set of rectangles T^{**}
remove T' from T^*, add all rectangles from T^{**} to T^*
add every rectangle of T^* to *Rect* as supports
remove a random support from *Rect* until $|Rect| > N$
while $|Rect| < N + K$ **do** ▷ holes generating
set $l_{hole} \in [mH, MH]$ and $w_{hole} \in [mH, MH]$ by random with step *TW*
set $x_{hole} \in [0, L - l_{hole}]$ and $y_{hole} \in [0, W - w_{hole}]$ by random with step *TW*
$Hole_{new} \leftarrow <(x_{hole}, y_{hole}), (l_{hole}, w_{hole})>$ ▷ new generated rectangle
check if any rectangle from *Rect* intersects with $Hole_{new}$ then **Continue**
return *Rect* ▷ set of supports and holes following geometric assumptions

Algorithm 15. Generator of experimental support plans with holes

The used version of the proposed algorithm does not consider the alignment problem mentioned in Section 3.2. The Voronoi diagrams built in the experiment violate the uniqueness restriction from Figure 1 of supports with west alignment. The approach of flipping the abscissa around the slab center is impractical due to the time consumptions and limitations.

The experiment runs each test three times per plan. Table 6 shows the average time and standard deviation for EP&DTL without holes, the proposed Voronoi diagram algorithm, and span determination algorithm from previous research [12] with minimum span area $min_S = 2$ m^2. Table 7 shows the ratio between average times for algorithms ("+" Voronoi faster, "–" Span faster). Table 8 shows the peak memory usage.

For 25 supports the proposed algorithm executes from 1.06 times faster without holes to 1.75 times slower with 56% holes than the span algorithm. For 50 supports the proposed algorithm executes from 1.04 times faster (for plans with 32% holes) to 2.36 times faster (for plans without holes) than the span algorithm, but to 1.07 times slower for plans with more holes. The proposed algorithm executes from 1.33 times faster (for 75 supports with 56% holes) to 15.17 times faster (for 200 supports without holes) than the span algorithm. The standard deviation of the time consumptions for the proposed algorithm is lower than for the span algorithm for every case. Holes accelerate the span algorithm to some extent, because the number of the decomposition is decreasing; holes slow down the Voronoi diagram algorithm as they create a more complex shadow structure. The memory usage has a near linear dependency on the number of supports and holes.

Table 6. Average time (in ms) and standard deviation of time consumptions

N	**25**		**50**		**75**		**100**	
K	**Voronoi**	Span	**Voronoi**	Span	**Voronoi**	Span	**Voronoi**	Span
0%	3.35	3.54	7.32	17.3	11.9	47.1	16.4	96.7
	.0002	.0010	.0003	.0037	.0004	.0086	.0005	.0164
8%	4.62	3.68	10.5	16.4	17.5	41.7	24.6	83.5
	.0004	.0011	.0008	.0036	.0009	.0073	.0023	.0137
20%	6.33	4.33	15.36	17.6	23.7	44.4	35.0	85.1
	.0005	.0012	.0009	.0035	.0016	.0073	.0024	.0121
32%	8.05	5.27	19.4	20.3	32.0	50.1	45.1	96.7
	.0005	.0014	.0011	.0037	.0022	.0072	.0031	.0123
44%	9.63	6.04	23.9	22.5	40.2	53.9	56.3	103
	.0007	.0014	.0014	.0039	.0027	.0074	.0034	.0125
56%	12.91	7.38	29.1	27.2	47.6	63.3	69.0	118
	.0008	.0016	.0020	.0045	.0033	.0084	.0045	.0134
N	**125**		**150**		**175**		**200**	
K	**Voronoi**	Span	**Voronoi**	Span	**Voronoi**	Span	**Voronoi**	Span
0%	20.8	169	26.5	267	31.1	400	37.2	565
	.0007	.0263	.0008	.0409	0.0008	.0617	.0009	.0812
8%	33.4	143	40.9	225	49.5	330	58.2	474
	.0026	.0224	.0032	.0330	.0034	.0495	.0039	.0702
20%	45.5	149	55.9	229	69.3	337	80.7	480
	.0031	.0205	.0034	.0302	.0042	.0438	.0047	.0617
32%	59.7	165	73.2	256	88.6	371	101	524
	.0033	.0209	0.0037	.0309	.0055	.0462	.0062	.0607
44%	73.8	173	90.7	263	106	401	123	552
	.0042	.0193	.0047	.0283	.0063	.0499	.0066	.0560
56%	88.3	197	106	300	126	448	147	616
	.0054	.0211	.0058	.0310	.0069	.0467	.0072	.0558

Table 7. Average time ratio comparison of the Voronoi and Span algorithms

	25	**50**	**75**	**100**	**125**	**150**	**175**	**200**
0%	+1.06	+2.36	+3.97	+5.88	+8.14	+10.1	+12.8	+15.2
8%	–1.26	+1.56	+2.38	+3.39	+4.29	+5.49	+6.66	+8.15
20%	–1.46	+1.15	+1.87	+2.43	+3.27	+4.11	+4.86	+5.94
32%	–1.53	+1.04	+1.57	+2.14	+2.77	+3.50	+4.19	+5.18
44%	–1.59	–1.06	+1.34	+1.83	+2.34	+2.90	+3.79	+4.50
56%	–1.75	–1.07	+1.33	+1.71	+2.23	+2.83	+3.56	+4.20

Table 8. Peak memory usage of the proposed algorithm for every set (in MB)

	25	**50**	**75**	**100**	**125**	**150**	**175**	**200**
0%	1,88	3,71	5,60	7,53	9,55	11,61	13,67	15,64
8%	2,49	4,71	7,26	9,60	12,24	14,82	17,33	20,02
20%	3,39	6,48	9,40	12,92	16,49	19,37	22,81	26,55
32%	3,92	7,88	12,07	15,98	20,35	24,41	28,65	33,05
44%	4,83	9,46	14,21	19,45	24,08	29,16	34,50	39,18
56%	5,57	11,13	16,64	22,50	28,31	34,43	39,88	46,09

The proposed algorithm can accelerate the comparative deformation process for a larger number of elements, while for the plans with fewer supports and more holes the span determination algorithm executes faster.

Figure 11 illustrates statistical values of the time consumption for the proposed algorithm. The gap of time consumption increases when the number of elements increases. The proposed holes processing always executes slower on average in comparison to without holes, even for the best case with holes and the worst case without them, except for the pair 25-0% and 25-8%. But the paired t-test for cases with and without holes showed that the difference in time execution cannot be explained only by the count of holes. Therefore, to understand the nature of the complexity brought by holes processing, several additional tests have taken place.

Figure 12 shows the average percent of plan processing time spent on event types. Spike event type also contains Blind Spike events. If some event type does not happen during the processing of some plan, the averaging function does not count it. Figure 13 shows the average time spent on one event processing. The plots do not show Inner Wave and Turn events, because they have small time consumptions: the Inner Wave event accounts for 0.96% and 0.024 ms, the Turn event accounts for 0.10% and 0.02 ms on average across all plans.

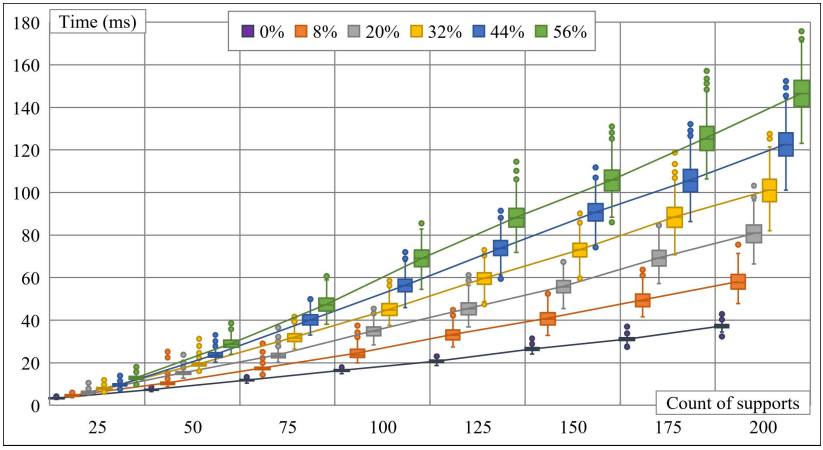


Fig. 11. Box plots for time consumption of the EP&DTL and proposed algorithms

The most time-consuming event by the average ratio to the whole process time is the West Edge event, taking from 25% to 50% of execution time. As the hole ratio grows, the ratio of hole events increases: from 17% for 8% plans to 52% for 56% plans. The most time-consuming hole event is the Begin Hole event. The average percentage of execution time for each event type does not vary by more than 1% with respect to the number of supports.

The average time for one execution of any type of event increases when the number of supports or holes increases. Among the average single execution times, the most time-consuming is the Begin Hole event. Four more events execute on average longer than 0.07 ms: West Edge, End Hole, East Corner and Outside Spike events. But East Corner and Outside Spike events take on average less than 4% and 3% of general time for any group. Therefore, let us focus on the research of the Begin Hole, West Edge and End Hole events for the further algorithm acceleration.

Figure 14 shows the average time spent on one function processing. Symbol "*" indicates the complex functions that include other simple functions. Function "Insert node" refers to the node insertion $Insert_{type}(\mathcal{T}, \not{b}v)$; function "Insert shadow" refers to the shadow insertion $Insert(\mathcal{T}, \beta v_1, \beta v_2)$; function "Fixate between" refers to the full execution of Algorithm 6. Functions with negligible execution times are not shown in the plot; the average single execution times for them across all plans are: Relocate is 0.0012 ms; Voronoi from Algorithm 2 is 0.0027 ms; Inserting a new event into $\mathcal{Q}$ is 0.0021 ms; Front is 0.0003 ms.

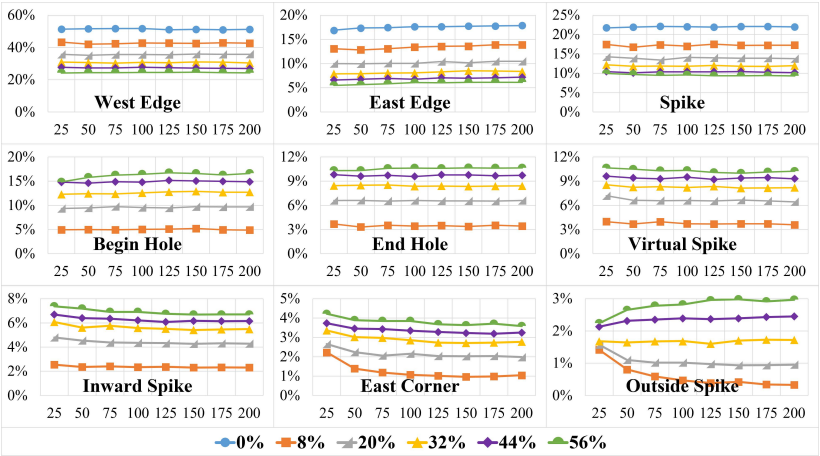


Fig. 12. Plots for processing rate of events; abscissa axis – count of supports; ordinate axis – average percent of processing time; series – ratio of holes number

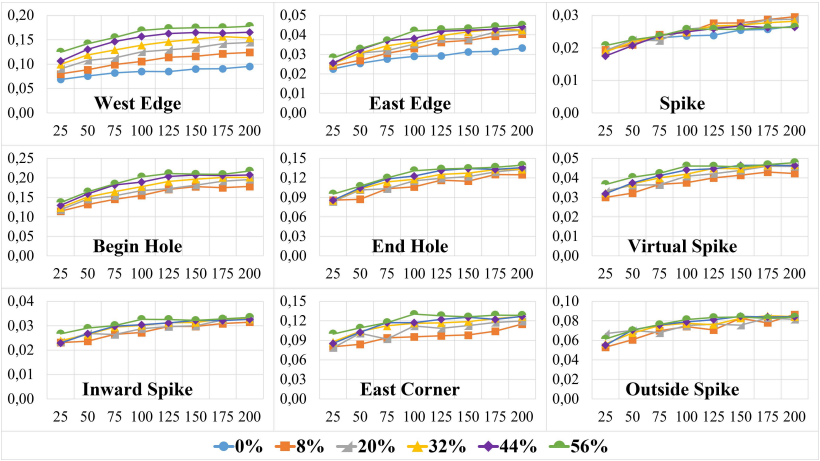


Fig. 13. Plots for time consumption of events; abscissa axis – count of supports; ordinate axis – average time of single execution in ms; series – ratio of holes number

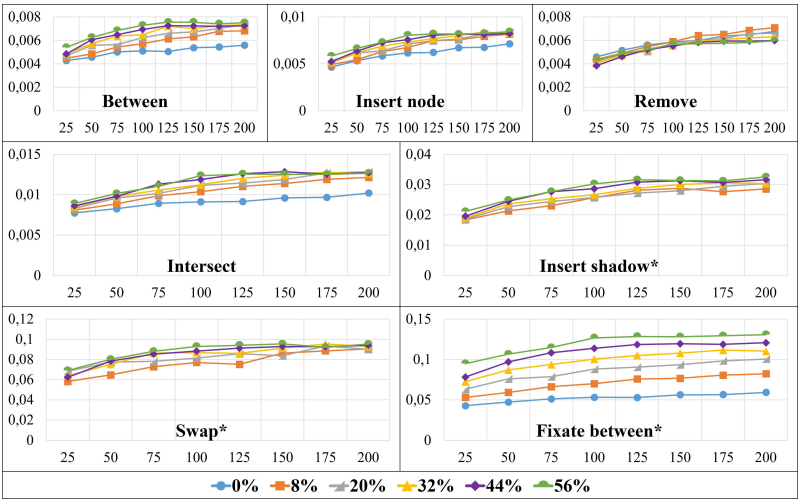


Fig. 14. Plots for time consumption of functions; abscissa axis – count of supports; ordinate axis – average time of single execution in ms; series – ratio of holes number

The most time-consuming function on average is the Fixate between from Algorithm 6. Function Swap executes slower for the 8% hole set for almost every number of supports. For other cases Fixate executes slower and has a clear correlation between the execution time and the number of elements, supports and holes; other functions do not have such an obvious

dependency. For all functions time consumption grows with the number of holes increasing, except Remove. Time consumption of Remove function on average decreases when the number of holes increases. This behavior is caused by the shadow structure: the number of simple removes from small shadow trees increases compared to removes from a single big tree.

Figure 15 shows the percentage of the time spent on operations in the execution of West Edge, Begin Hole and End Hole events. Operation "Voronoi" refers to the Algorithm 2, operation "Insert event" refers to inserting a new event into Q.

The most time-consuming operation in big events is Insert. Insert and Remove operations spend around 40% of West Edge, 45% of Begin Hole and 70% of End Hole events. The percentage of the Remove operation increases when the number of elements increases. For the Begin Hole and End Hole events the percentage of the Insert operation increases when the number of holes increases. For the West Edge event, the percentage of the Insert operation increases when the number of supports increases and the number of holes decreases, because not every shadow during fixation of the wavefront produces new nodes.

Though Insert and Remove operations account for most of the execution time in big events, they do not have much optimization potential, as they are already well-optimized operations of a standard AVL-tree.

Between and Relocate operations do not account for much execution time of big events. The Intersect operation in West Edge and Begin Hole events does not seem to be optimizable, as it uses a common mechanism of the AVL tree and may become less time-consuming only with another structure of the shadows. The Voronoi operation from Algorithm 2 accounts for 15% of West Edge event time, but it has a low average execution time of 0.0027 ms. The insertion of a new event into Q accounts for around 5% of West Edge, 17% of Begin Hole and 25% of End Hole events, though the average time of a single operation is very low (around 0.0021 ms). The main reason is the insertion of new events for reinserted nodes from the between sets during Begin Hole and End Hole events (steps 6, 12, 19, 20 in Algorithm 7; step 9 in Algorithm 8). Then a possible optimization can lie in searching for new events only for the first and last nodes in these sets, reassigning the existing events for every other node.

Besides, the experimental program implements the priority queue Q as a list with insertion complexity O(N), N is the number of events in Q, and extraction complexity O(1). The binary heap implementation with insertion and extraction complexity O(logN) can optimize this operation.

Thus, the main optimization to decrease the time consumption of the proposed algorithm lies in a better implementation of the event

creation operation, especially during the processing of the between sets. Other optimizations can lie in restructuring the shadow trees.

The experiment showed that the proposed algorithm is faster than the previous span determination algorithm for plans with more supports and fewer holes: 1.06 times faster for plans with 25 supports without holes and 1.75 times slower with 14 holes (56% of supports); 2.36 times faster with 50 supports without holes and 1.07 times slower with 28 holes; with 75 supports or more, the algorithm is from 1.33 times faster (for 75 supports and 42 holes) to 15.2 times faster (for 200 supports without holes). Thus, the proposed algorithm extends the scope of the deformation comparative analysis for plans with a large number of supports.

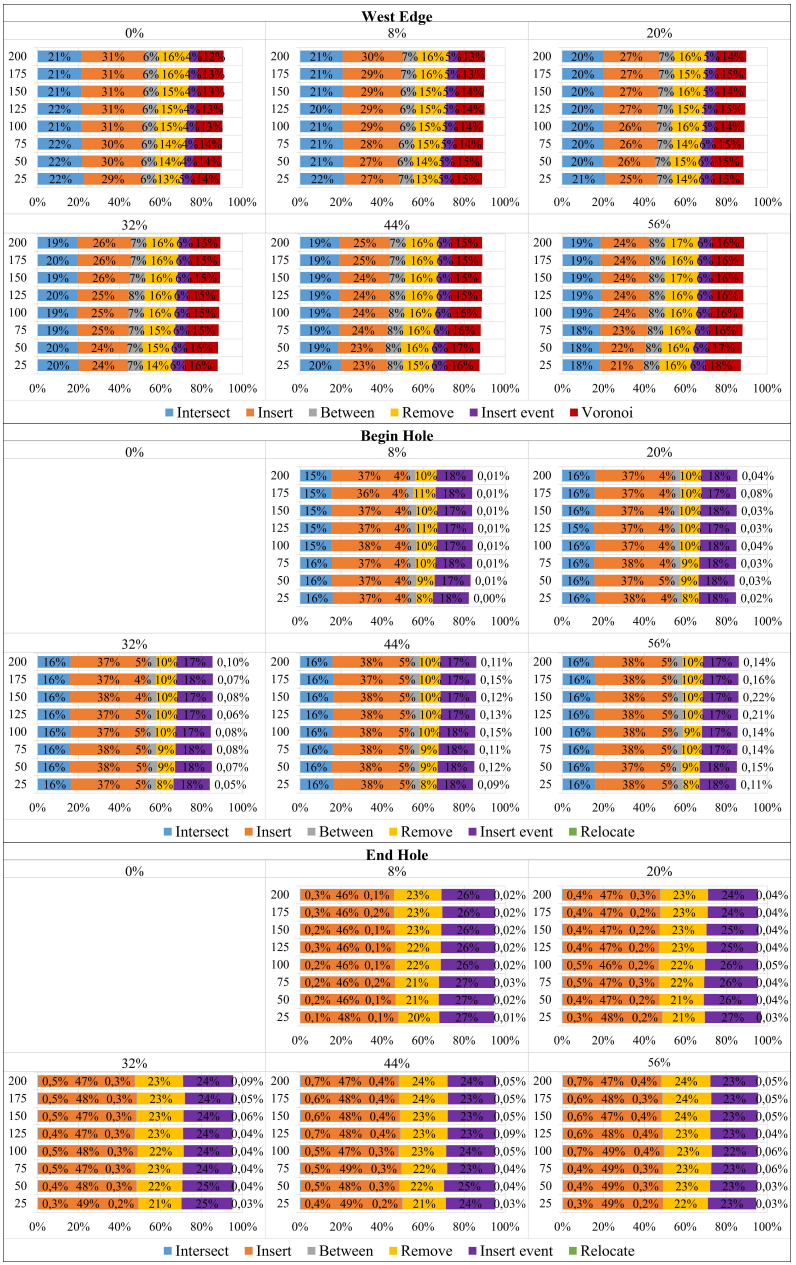


Fig. 15. Functions percentages in processing of the biggest events

5.2. Correlation analysis preliminary experiment. This section describes a preliminary test of the Voronoi diagram usage for deformation analysis. The experiment estimates correlations between the Voronoi cells and deformation, evaluated by the finite element (FE) method, in the slab. The aim of the experiment is to give a first check of the algorithm in the deformation comparative analysis and set a direction for future research.

Table 9 describes the sets of deformation metrics, which were chosen for the preliminary correlation analysis, Voronoi parameters and simple parameters of the supports for comparison.

Table 9. Voronoi cells parameters, simple support parameters and deformation metrics for preliminary correlation analysis

Deformation metrics	
Mx	The average value of the bending moment perpendicular to the X axis
My	The average value of the bending moment perpendicular to the Y axis
Z	The difference between extreme vertical deflections of vertices
Reinf	The amount of reinforcement needed in every finite element
Voronoi parameters	
Area	Area of the Voronoi cell
P	Perimeter of the Voronoi cell
CD	Distance between center of the Voronoi cell and center of the support
MD	Maximum distance between the support and edges of the Voronoi cell
AD	Average distance between the support and edges of the Voronoi cell
LX	The maximum difference in vertices abscissa of the Voronoi cell
LY	The maximum difference in vertices ordinate of the Voronoi cell
Simple support parameters	
SL	Length of the support
SDE	Distance from the support edge to the edge of the slab
SDS	Distance to the closest support
SDH	Distance to the closest hole
SLS	Density of supports in the local area around the support (relatively to local area)
SLH	Density of holes in the local area around the support (relatively to local area)

To handle the alignment problem mentioned in Section 3.2, the experiment applies the approach of flipping the abscissa approach with some secondary processing mechanisms. The following modifications define the approach:

1) To process simultaneous West Edge and East Corner events, we use a Fixation event instead; instead of the original events, we create Fixation events; each Fixation event has a flag indicating the original event type and executes as the original would; Fixation events have the same order position among events between East Corner and Begin Hole events;

2) The approach builds the Voronoi diagram twice: first with the abscissa of the elements flipped about the slab center, second with the original abscissa; after the second diagram is built, cells get edges from the

flipped copy, that do not appear in the original diagram; this yields the correct Voronoi edges according to the uniqueness rule from Section 3.2 for aligned supports;

3) A Fixation event does not produce Voronoi edges between nodes during Fixate operations (Algorithm 6: steps 12, 13 and 27) for aligned supports; thus, both diagrams do not have incorrect edges;

4) Every Voronoi edge should have the maximum possible length, meaning both original and flipped diagrams have only necessary vertices;

5) If there are supports with east alignment that have a Voronoi edge with supports that have west alignment on the east side of their cells, then disjoint vertices appear; each of them should be matched with the closest pair among other disjoint vertices in the cell with the same abscissa;

6) The Fixation operation can create bisectors with incorrect assignment of the owner for aligned supports in the processing of inner shadows (Fig. 7c, Algorithm 6: steps 24 and 25, T_i); therefore, there should be a check operation that will give the correct owner for these assignments in the case of support alignment during fixation; it can be done by creating polygonal zones for every valid Fixation event, such as $Zone = <\{p_1, p_2, \ldots, p_{NZ}\}, T_{Zone}>$ by set of points, NZ is the count of points in the polygon, and the support $T_{Zone} \in T$.

Algorithm 16 describes an operation to find zones of assignment before the first simultaneous Fixation event.

To clip a new zone by a simple convex polygon, the Sutherland–Hodgman algorithm can be used. To clip the previous zones by the new one, the new zone can be split into convex parts. During the fixation operation we need to find a zone that contains the beginning point for every new bisector (Algorithm 6: steps 25, 32 and 34), for example by the ray casting algorithm, and set the corresponding support as an owner in the *Create* function for it. Then the described approach will give the correct ownership for every created bisector, even for simultaneous Fixation events inside the inner shadows.

The flipping abscissa approach handles the alignment problem, but is time-consuming and should be used only for experimental purposes. A more convenient and practical approach is a subject of future research.

The FE mesh is overlaid onto the Voronoi diagram. If some finite element or vertex of the mesh lies inside several Voronoi cells, we apply decision rules. For vertices:

- find the Voronoi cell with the closest support to the vertex;
- if there are several such cells, then find the Voronoi cell with the closest centroid to the vertex;
- if there are several such cells, then the vertex is ignored.

For finite elements:

- find the Voronoi cell that contains the most vertices of the finite element;
- if there are several such cells, then find the Voronoi cell with the closest centroid to the center of the finite element;
- if there are several such cells, then the finite element is ignored.

Input $\mathcal{T} \neq \varnothing$, $q = Q.pop()$, $\mathcal{L} = Priority(q)$, $type(q) = $ Fixation
1. $Q^*_{Sim} \leftarrow \{q' \in Q \mid Priority(q') = \mathcal{L}, type(q') = \text{Fixation}\}$ ▷ simultaneous fixation events
2. order Q^*_{Sim} by ascending y, then by descending x
3. $Holes^* \leftarrow \{Hole_k \in Holes \mid x_k < \mathcal{L}\}$ ▷ holes on the left of sweep line
4. order $Holes^*$ by descending x, then by ascending y
5. $q_{Last} = \varnothing$, $Zones \leftarrow \{\varnothing\}$ ▷ polygonal zones of assignment
6. **for** $q_{New} \in Q^*_{Sim}$ **do**
7. set T_{New} as support of q_{New} event, set y_{New} as ordinate of T_{New}, set w_{New} as width of T_{New}
8. set $Zone_{Last}$ as $Zones$.Last() if $Zones \neq \varnothing$
9. **if** q_{New} is West Edge Fixation **then**
10. $Zone_{New} \leftarrow <\{(0, y_{New} + w_{New} + \mathcal{L}), (\mathcal{L}, y_{New} + w_{New}), (\mathcal{L}, y_{New}), (0, y_{New} - \mathcal{L})\}, T_{New}>$
11. **if** $q_{Last} \neq \varnothing$ **and** q_{Last} is West Edge Fixation **and** $Zone_{Last} \cap Zone_{New} \neq \varnothing$ **then**
12. ▷ find the rightest point of zones intersection
13. set $p_{Inter} \in Zone_{Last} \cap Zone_{New}$: $\nexists p' \in Zone_{Last} \cap Zone_{New}$, $p' \neq p_{Inter}$, $x' > x_{Inter}$
14. $Cut_{New} \leftarrow \{(0, -\mathcal{L}), (0, y_{Inter}), (\mathcal{L}, y_{Inter}), (\mathcal{L}, -\mathcal{L})\}$ ▷ cutting area for new zone
15. $Zone_{New} \leftarrow Zone_{New} \setminus Cut_{New}$ ▷ cutting zone by mid line
16. **else** set $Hole_k$ as hole of Q_{New} event, set $v_k{}^e$ as corresponded east corner of hole
17. set (f, w) as $(-1, 0)$ if $v_k{}^e$ is south corner or as $(1, w_k)$ otherwise
18. $Zone_{New} \leftarrow <\{v_k{}^e, (x_k, y_k + w), (0, y_k + w - f \cdot x_k), (0, y_k + w + f \cdot (x_k + l_k))\}, T_{New}>$
19. **for** $Hole' \in Holes^*$ **do**
20. check if west edge of $Hole'$ is passed by wavefront by $Inner = \varnothing$ then **Continue**
21. check if $Zone_{New} \cap Hole' = \varnothing$ then **Continue**
22. $v'^s \leftarrow (x' + l', y')$, $v'^n \leftarrow (x' + l', y' + w')$ ▷ east corners of $Hole'$
23. **if** $v'^s \notin Zone_{New}$ **and** $v'^n \notin Zone_{New}$ **then** ▷ zone is locked by hole
24. $Cut_{Hole} \leftarrow \{(0, -\mathcal{L}), (0, W + \mathcal{L}), (x' + l', W + \mathcal{L}), (x' + l', -\mathcal{L})\}$
25. $Zone_{New} \leftarrow Zone_{New} \setminus Cut_{Hole}$, **Break**
26. **else if** $v'^s \in Zone_{New}$ **and** $v'^n \in Zone_{New}$ **then** ▷ zone includes hole
27. find q'^s and q'^n in Q^*_{Sim} with East Corner flag related to v'^s and v'^n if exist
28. remove zones from $Zones$ created by q'^s and q'^n if exist
29. remove q'^s and q'^n from Q^*_{Sim} if exist
30. **else if** $v'^s \in Zone_{New}$ **then**
31. $Zone_{New} \leftarrow Zone_{New} \setminus \{v'^s, (x', y'), (0, y' + x'), (0, W + \mathcal{L}), (x' + l', W + \mathcal{L})\}$
32. find q'^s in Q^*_{Sim} with East Corner flag related to v'^s if exists
33. remove zone from $Zones$ created by q'^s, remove q'^s from Q^*_{Sim} if exists
34. **else** $Zone_{New} \leftarrow Zone_{New} \setminus \{v'^n, (x', y' + w'), (0, y' - x'), (0, -\mathcal{L}), (x' + l', -\mathcal{L})\}$
35. **if** $\exists Zone' \in Zones$: $v'^n \in Zone'$ **then** ▷ need to cut by mid of hole
36. $Zone_{New} \leftarrow Zone_{New} \setminus \{(x', y' + w'/2), (0, y' + w'/2), (0, -\mathcal{L}), (x', -\mathcal{L})\}$

find q'^n in Q^*_{Sim} with East Corner flag related to v'^n if exists
remove zone from *Zones* created by q'^n, remove q'^s from Q^*_{Sim} if exists
$\forall Zone' \in Zones: Zone' \leftarrow Zone' \setminus Zone_{\text{New}}$
return *Zones* ▷ assignment zones intersecting only by edges

Algorithm 16. Creation of assignment zones for inner shadows fixation

The deformations are evaluated using LIRA-FEM on a plane slab; the mesh step is 125 mm. The slab settings: width is 250 mm; elastic modulus is 3e+006 t/m^2; Poison's ratio is 0.2; specific gravity is 2.502 t/m^3 (B25). The reinforcement setting: A500 for longitudinal reinforcement and A240 for crosswise reinforcement. The slab is modeled by the plate finite elements. The slab contains uniformly distributed orthogonal loads equal to 0.6255 t/m^2. Supports are modeled as restraints along the axis orthogonal to the slab plane (Z axis). The technical setup is the same as in the previous experiment.

The experiment uses 250 plans with 25 supports and with hole ratios from 0% to 56% from the previous experiment, for a total of 1500 plans, computing correlations for a set of the constructed Voronoi cells. Thus, the sample size is 37500 Voronoi cells in total, with 6250 cells per hole percentage group. The local offset for SLS and SLH simple parameters equals 0.5 m; the local area is a rectangle with edges formed by offset from the support edges.

Table 10 and Table 11 show paired Pearson's correlation coefficients between Voronoi and support simple parameters and deformation metrics. The correlation values for SDH and SLH do not include plans without holes; therefore, their sample size is 31250 instead of 37500. To counteract the multiple comparisons problem, the experiment applies the Bonferroni correction with a significance level of 0.01 for 52 tests; the adjusted significance level is approximately 0.00019. The tables include p-value and confidence interval for coefficients using the corrected significance level.

The result of the experiment shows the existence of a correlation between the parameters of Voronoi diagram and deformation values. The bending moments (Mx, My) have moderate (>.3) correlation strength with the maximum distance to Voronoi vertex parameter (MD) and weak (>.1) with others, except the pairs LX-Average Mx and LY-Average My, that have negligible correlation coefficients (<0.1). This effect can be explained by the bending moment Mx acting around the x-axis; consequently, the larger the dimension of the slab in the orthogonal direction along the y-axis, the larger the bending moment Mx will be. The same reasoning applies to My. The range of the vertical deflection (Z) has strong (>.5) correlation coefficients with the area of the Voronoi cell (Area), distance between the center of the Voronoi cell and the center of the support (CD), maximum and average distance to Voronoi vertices from the support (MD, AD) and

moderate correlation coefficients with others. The required reinforcement amount (Reinf) has a strong correlation with all Voronoi parameters, the strongest being with Area: 0.763.

Table 10. Correlations between Voronoi parameters and deformation metrics

	Average Mx	**Average My**	**Max – min Z**	**Sum Reinf**
Area	*r* = .195 *p* = .00000 CI [.176 .213]	*r* = .231 *p* = .00000 CI [.213 .249]	*r* = .516 *p* = .00000 CI [.502 .530]	*r* = .763 *p* = .00000 CI [.755 .771]
P	*r* = .187 *p* = .00000 CI [.168 .205]	*r* = .211 *p* = .00000 CI [.193 .229]	*r* = .482 *p* = .00000 CI [.467 .496]	*r* = .652 *p* = .00000 CI [.641 .663]
CD	*r* = .241 *p* = .00000 CI [.223 .259]	*r* = .263 *p* = .00000 CI [.245 .281]	*r* = .581 *p* = .00000 CI [.568 .593]	*r* = .550 *p* = .00000 CI [.537 .563]
MD	*r* = .334 *p* = .00000 CI [.317 .351]	*r* = .346 *p* = .00000 CI [.329 .363]	*r* = .682 *p* = .00000 CI [.671 .692]	*r* = .717 *p* = .00000 CI [.707 .726]
AD	*r* = .285 *p* = .00000 CI [.267 .303]	*r* = .305 *p* = .00000 CI [.288 .323]	*r* = .613 *p* = .00000 CI [.601 .625]	*r* = .718 *p* = .00000 CI [.708 .727]
LX	*r* = .039 *p* = .00000 CI [.020 .058]	*r* = .294 *p* = .00000 CI [.277 .312]	*r* = .404 *p* = .00000 CI [.388 .420]	*r* = .565 *p* = .00000 CI [.552 .578]
LY	*r* = .292 *p* = .00000 CI [.275 .310]	*r* = .078 *p* = .00000 CI [.059 .097]	*r* = .435 *p* = .00000 CI [.419 .450]	*r* = .571 *p* = .00000 CI [.558 .584]

Table 11. Correlations between simple support parameters and deformation metrics

	Average Mx	**Average My**	**Max – min Z**	**Sum Reinf**
SL	*r* = –.003 *p* = .53049 CI [–.022 .016]	*r* = .002 *p* = .75768 CI [–.018 .021]	*r* = .096 *p* = .00000 CI [.077 .115]	*r* = .254 *p* = .00000 CI [.236 .272]
SDE	*r* = –.021 *p* = .00004 CI [–.041 –.002]	*r* = –.025 *p* = .00000 CI [–.044 –.006]	*r* = .015 *p* = .00424 CI [–.004 .034]	*r* = .010 *p* = .04674 CI [–.009 .030]
SDS	*r* = –.018 *p* = .00068 CI [–.037 .002]	*r* = .015 *p* = .00466 CI [–.005 .034]	*r* = .137 *p* = .00000 CI [.118 .156]	*r* = .273 *p* = .00000 CI [.255 .291]
SDH	*r* = –.031 *p* = .00000 CI [–.052 –.010]	*r* = –.009 *p* = .11890 CI [–.030 .012]	*r* = –.061 *p* = .00000 CI [–.082 –.040]	*r* = –.068 *p* = .00000 CI [–.089 –.047]
SLS	*r* = –.111 *p* = .00000 CI [–.130 –.092]	*r* = –.101 *p* = .00000 CI [–.120 –.082]	*r* = –.147 *p* = .00000 CI [–.165 –.128]	*r* = –.221 *p* = .00000 CI [–.239 –.203]
SLH	*r* = .056 *p* = .00000 CI [.035 .077]	*r* = .033 *p* = .00000 CI [.012 .054]	*r* = .098 *p* = .00000 CI [.077 .118]	*r* = .084 *p* = .00000 CI [.063 .105]

Only SLS and SLH simple support parameters have significant correlation coefficients with each deformation metric, while the others have some insignificant coefficients. The parameter of distance to the edge of the slab (SDE) has only one significant but negligible coefficient with Average

My and has insignificant coefficients with others; thus, it seems that this parameter does not bring any important information to the analysis. The parameters of distance to the closest hole (SDH) and the density of holes in the local area (SLH) have significant but negligible coefficients, except of the pair SDH-Average My, meaning these simple heuristics are not enough to describe deformations.

The length of the support parameter (SL) has a significant weak correlation with reinforcement needed, a significant but negligible correlation with deflection difference, and insignificant coefficients with other deformation metrics. The parameter of distance to the closest support (SDS) has a significant weak correlation coefficient with the difference of Z deflections and the needed reinforcement amount, and it has insignificant coefficients with the average bending moments. The parameter of the density of supports in the local area (SLS) has significant weak correlation coefficients with all deformation metrics.

The following conclusions are made based on the correlation analysis results:

1) Area and perimeter (P) of Voronoi cells have an obvious correlation with deformations in the slab: the bigger Voronoi cell leads to bigger spans with neighboring supports, which leads to greater deformation in the slab;

2) The distance between the center of the Voronoi cell and the center of the support (CD) describes the imbalance of the supporting area shape in the slab, leading to bigger bending moments and deflections;

3) The maximum and average distance to the Voronoi cell vertices (MD and AD) describe the length of the maximum and average span from the support around it; the larger the span length, the greater the deformations occur;

4) The bounding box dimensions in abscissa and ordinate of the Voronoi cell (LX and LY) should explain the bending moments, because the larger the extent of the supporting area, the larger the bending moments should be; but it seems that the distance to the Voronoi vertices from the support (MD, AD) describes deformations of the slab better;

5) Some simple parameters, such as length of the support (SL), distance to the closest support (SDS) and density of supports in the local area (SLS) have a significant weak correlation and can be used in deeper analysis, but for any deformation metric all of them have less correlation power than the Voronoi parameters.

In general, the Voronoi diagram seems to be a useful heuristic approach to evaluate slab deformation, because its nature reflected in the parameters of the Voronoi cells has a significant and valuable correlation

with the important deformation metrics. Future research will use the proposed algorithm and the obtained results of the preliminary correlation analysis as a starting point. Besides, the limitations of the described model, such as the single value for dimensions and thickness of the model slab, the single type of concrete and reinforcement tested, should be researched further as well to confirm the applicability of the Voronoi diagram to a wider range of floor slab and building design variations.

6. Conclusion. A new Voronoi diagram algorithm with rectangular sites and holes inside has been proposed. The new algorithm uses the algorithm of E. Papadopoulou and D.T. Lee [10] (EP&DTL) as the base and can process the rectangular sites fast enough by implementing the L_∞ distance metric and straightforward sweep line events. Therefore, the Voronoi diagram can be applied to the slab deformation analysis problem as part of the comparative deformation analysis research, as in [19] and [20].

The modification of the EP&DTL algorithm with rectangular holes inside the Voronoi diagram, which correspond to the holes in the floor slab, has been described. The new structure in the height-balanced binary tree representing the wavefront called "shadows" restricts the intersection operation during the sweep line algorithm. New "virtual" bisectors induced by holes maintain "shadow" structures, while the new events corresponding to them allow handling holes during the Voronoi algorithm.

The computational experiment of comparing time consumption against the previous span determination approach [19] has demonstrated that the proposed algorithm executes from 1.33 times faster for the plans with 75 supports and 42 holes to 15.17 times faster for the plans with 200 supports without holes. But the proposed algorithm executes slower for smaller cases, especially with more holes, up to 1.75 times slower for 25 supports and 14 holes. The extended analysis of the time consumption for events and functions has shown some potential optimizations.

The preliminary correlation analysis of the Voronoi cell parameters has shown a significant strong correlation between the reinforcement needed and the area of the cell (correlation coefficient 0.76) and the distance to the vertices (correlation coefficient 0.72). However, these findings should be interpreted with caution due to the simplified assumptions of the model, while the algorithm has a limitation in processing support alignment. Therefore, future research will propose a solution for the alignment problem and will use the obtained results to extend the analysis to build a practical slab deformation heuristic evaluation using the Voronoi diagram approach.

References

1. Hadi M.-H., Sharafi P., Teh L.-H. A new formulation for the geometric layout optimisation of flat slab floor systems. Proceeding of the Australasian Structural Engineering Conference: The Past, Present and Future of Structural Engineering. Perth, Western Australia: Engineers Australia. 2012. pp. 122–129. DOI: 10.3316/informit.026275045488938.
2. Petprakob W. Beam-slab floor optimization using genetic and particle swarm optimization algorithms. A Thesis for degree of master of science in engineering and technology. Thailand: Thammasat University, Sirindhorn International Institute of Technology. 2014. 90 p. DOI: 10.14457/TU.the.2014.490.
3. Beeby A.-W., Narayanan R.-S. Designers' Guide to EN 1992-1-1 and EN 1992-1-2. Eurocode 2: Design of Concrete Structures Design of Concrete Structures. General Rules and Rules for Buildings and Structural Fire Design. London: Thomas Telford Publishing, 2005. 230 p. DOI: 10.1680/dgte2docs.31050.
4. Sahab M.-G., Ashour A.-F., Toropov V.-V. Cost optimisation of reinforced concrete flat slab buildings. Engineering Structures. 2005. vol. 27. no. 3. pp. 313–322. DOI: 10.1016/j.engstruct.2004.1.002.
5. Sharafi P. Cost optimization of the preliminary design layout of reinforced concrete framed buildings. A Thesis for degree of Doctor of Philosophy in civil engineering. Australia: University of Wollongong, School of Civil, Mining and Environmental Engineering. 2013. 286 p.
6. Meng X., Lee T.-U., Xiong Y., Huang X., Xie Y.-M. Optimizing support locations in the roof – column structural system. Applied Science. 2021. vol. 11. no. 6. 2775 p. DOI: 10.3390/app11062775.
7. Zelickman Y., Amir O. Optimization of column layouts in buildings considering structural and architectural constraints. Engineering Archive (engrXiv). 2024. 45 p. DOI: 10.31224/2723.
8. Leshkevich O. [The use of artificial neural networks to evaluate the reinforcement of reinforced concrete floor slabs. Problems of modern concrete and reinforced concrete: Collection of scientific papers no. 11.]. Minsk: In-t BelNIIS, 2019. pp. 51–62. DOI: 10.35579/2076-6033-2019-11-04. (In Russ.)
9. Wang J., Chen K., Yang H., Zhang L. Ensemble deep learning enabled multi-condition generative design of aerial building machine considering uncertainties. Automation in Construction. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105134.
10. Liao W., Lu X., Fei Y., Gu Y., Huang Y. Generative AI design for building structures. Automation in Construction. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105187.
11. Steiner B., Mousavian E., Mehdizadeh F.-S., Wimmer M., Musialski P. Integrated structural-architectural design for interactive planning. Computer Graphics Forum. 2016. vol. 36. no. 8. pp. 1–13. DOI: 10.1111/cgf.12996.
12. Zinov V., Kartak V., Valiakhmetova Yu. [Algorithm for assessing the deformation of floor slabs based on the building's span-support schemes]. Sistemy analiza i obrabotki dannykh – Analysis and Data Processing Systems. 2023. vol. 92. no. 4. pp. 35–54. DOI: 1.17212/2782-2001-2023-4-35-54. (In Russ.).
13. Zinov V., Kartak V., Valiakhmetova Yu. [Solving a multi-criteria problem of rational placement of load-bearing walls using a genetic algorithm]. Informatika i avtomatizatsiya – Informatics and Automation. 2025. vol. 24. no. 2. pp. 464–491. DOI: 1.15622/ia.24.2.4. (In Russ.).
14. Jung C., Redenbach C. Crack modeling via minimum-weight surfaces in 3d Voronoi diagrams. Journal of Mathematics in Industry. 2023. vol. 13. no. 10. DOI: 10.1186/s13362-023-00138-1.
15. Polat H., Ilerisoy Z.-Y. A geometric method on facade form design with Voronoi Diagram. Modular. 2020. vol. 3. no. 2. pp. 179–194.

16. Rokicki W., Gawell E. Voronoi diagrams – architectural and structural rod structure research model optimization. Malowsze Studia Regoinalne. 2016. vol. 19. pp. 155–164. DOI: 10.21858/msr.19.10.
17. Fortune S. A sweepline algorithm for Voronoi diagrams. Algorithmica. 1987. vol. 2. pp. 153–174. DOI: 10.1007/BF01840357.
18. Bhattacharya P., Gavrilova M.-L., et al. Voronoi diagram in optimal path planning. Proceeding of the 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD). IEEE. 2007. pp. 38–47. DOI: 10.1109/ISVD.2007.43.
19. Ho Y.-J., Liu J.-S. Smoothing Voronoi-based obstacle-avoiding path by length-minimizing composite Bezier curve. Proceeding of the International Conference on Service and Interactive Robotics (SIRCon09). Institute of Information Science: Academia Sinica. 2009.
20. Papadopoulou E., Lee D.-T. Critical area computation via Voronoi Diagrams. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1999. vol. 18. no. 4. pp. 463–474. DOI: 10.1109/43.752929.

Zinov Vladislav — Postgraduate student, Department of Technical Cybernetics, Federal State Budgetary Educational Institution of Higher Education «Ufa University of Science and Technology». Research interests: optimization in the field of building design, heuristic and metaheuristic methods, and decision support systems. The number of publications — 8. zinovvladislavufa@gmail.com; 32, Zaki Validi St., 450076, Ufa, Russia; office phone: +7(917)435-5560.

УДК 519.25, 624.042 **DOI** 10.15622/ia.25.2.9

В.И. Зинов

ОБРАБОТКА ОТВЕРСТИЙ В ДИАГРАММЕ ВОРОНОГО С ПРЯМОУГОЛЬНЫМИ ОСНОВАНИЯМИ ДЛЯ СРАВНИТЕЛЬНОГО АНАЛИЗА ДЕФОРМАЦИЙ В ПЛИТАХ

Зинов В.И. **Обработка отверстий в диаграмме Вороного с прямоугольными основаниями для сравнительного анализа деформаций в плитах.**

Аннотация. В данной статье предложен новый алгоритм построения диаграммы Вороного на прямоугольных основаниях с отверстиями. Алгоритм основан на алгоритме построения диаграммы Вороного с метрикой расстояния L_∞, разработанным Papadopoulou E. и Lee D.-T. Предложены модификации для обработки отверстий в модели диаграммы Вороного. Алгоритм обрабатывает искажения в структуре диаграммы, вызванные отверстиями, создавая слои береговой линии при построении диаграммы Вороного, названные тенями, и используя новый тип биссекторов, которые не задают ребер на диаграмме Вороного, но являются основой для структуры слоев береговой линии. Алгоритм задает новые события для заметающей прямой, сохраняя общие принципы обработки согласно базовому алгоритму. По результатам сравнения времени выполнения с предыдущим подходом определения пролетов в плите, предложенный алгоритм работает в 1.33 раза быстрее для плана с 75 опорами и в 15.17 раз быстрее для наибольшего протестированного количества опор, но медленнее для меньшего количества опор и большего числа отверстий. Предварительный корреляционный анализ показал наличие значимой линейной корреляции с коэффициентом 0.76 между площадью ячейки Вороного и требуемым количеством армирования, а также сильную и умеренную корреляцию между другими параметрами ячейки и метриками деформаций. В заключении указаны текущие ограничения модели и алгоритма, которые будут исследованы в дальнейшем.

Ключевые слова: диаграмма Вороного, сравнительная оценка деформаций, рациональное размещение опор, корреляционная модель, оптимизация проектирования зданий.

Литература

1. Hadi M.-H., Sharafi P., Teh L.-H. A new formulation for the geometric layout optimisation of flat slab floor systems // Proceeding of the Australasian Structural Engineering Conference: The Past, Present and Future of Structural Engineering. Perth, Western Australia: Engineers Australia. 2012. pp. 122–129. DOI: 10.3316/informit.026275045488938.
2. Petprakob W. Beam-slab floor optimization using genetic and particle swarm optimization algorithms // A Thesis for degree of master of science in engineering and technology. Thailand: Thammasat University, Sirindhorn International Institute of Technology. 2014. 90 p. DOI: 10.14457/TU.the.2014.490.
3. Beeby A.-W., Narayanan R.-S. Designers' Guide to EN 1992-1-1 and EN 1992-1-2. Eurocode 2: Design of Concrete Structures Design of Concrete Structures. General Rules and Rules for Buildings and Structural Fire Design // London: Thomas Telford Publishing, 2005. 230 p. DOI: 10.1680/dgte2docs.31050.
4. Sahab M.-G., Ashour A.-F., Toropov V.-V. Cost optimisation of reinforced concrete flat slab buildings // Engineering Structures. 2005. vol. 27. no. 3. pp. 313–322. DOI: 10.1016/j.engstruct.2004.1.002.

5. Sharafi P. Cost optimization of the preliminary design layout of reinforced concrete framed buildings // A Thesis for degree of Doctor of Philosophy in civil engineering. Australia: University of Wollongong, School of Civil, Mining and Environmental Engineering. 2013. 286 p.
6. Meng X., Lee T.-U., Xiong Y., Huang X., Xie Y.-M. Optimizing support locations in the roof – column structural system // Applied Science. 2021. vol. 11. no. 6. 2775 p. DOI: 10.3390/app11062775.
7. Zelickman Y., Amir O. Optimization of column layouts in buildings considering structural and architectural constraints // Engineering Archive (engrXiv). 2024. 45 p. DOI: 10.31224/2723.
8. Лешкевич О.Н. Использование искусственных нейронных сетей для оценки армирования железобетонных плит перекрытия. Проблемы современного бетона и железобетона: Сб. научн. тр. № 11. // Минск: Ин-т БелНИИС. 2019. С. 51–62. DOI: 10.35579/2076-6033-2019-11-04.
9. Wang J., Chen K., Yang H., Zhang L. Ensemble deep learning enabled multi-condition generative design of aerial building machine considering uncertainties // Automation in Construction. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105134.
10. Liao W., Lu X., Fei Y., Gu Y., Huang Y. Generative AI design for building structures // Automation in Construction. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105187.
11. Steiner B., Mousavian E., Mehdizadeh F.-S., Wimmer M., Musialski P. Integrated structural-architectural design for interactive planning // Computer Graphics Forum. 2016. vol. 36. no. 8. pp. 1–13. DOI: 10.1111/cgf.12996.
12. Зинов В.И., Картак В.М., Валиахметова Ю.И. Алгоритм оценки деформации плит перекрытий по пролетно-опорным схемам здания // Системы анализа и обработки данных. 2023. Т. 92. № 4. С. 35–54. DOI: 10.17212/2782-2001-2023-4-35-54.
13. Зинов В.И., Картак В.М., Валиахметова Ю.И. Решение многокритериальной задачи рационального размещения несущих стен с помощью генетического алгоритма // Информатика и автоматизация. 2025. Т. 24. № 2. С. 464–491. DOI: 10.15622/ia.24.2.4.
14. Jung C., Redenbach C. Crack modeling via minimum-weight surfaces in 3d Voronoi diagrams // Journal of Mathematics in Industry. 2023. vol. 13. no. 10. DOI: 10.1186/s13362-023-00138-1.
15. Polat H., Ilerisoy Z.-Y. A geometric method on facade form design with Voronoi Diagram // Modular. 2020. vol. 3. no. 2. pp. 179–194.
16. Rokicki W., Gawell E. Voronoi diagrams – architectural and structural rod structure research model optimization // Malowsze Studia Regoinalne. 2016. vol. 19. pp. 155–164. DOI: 10.21858/msr.19.10.
17. Fortune S. A sweepline algorithm for Voronoi diagrams // Algorithmica. 1987. vol. 2. pp. 153–174. DOI: 10.1007/BF01840357.
18. Bhattacharya P., Gavrilova M.-L., et al. Voronoi diagram in optimal path planning // Proceeding of the 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD). IEEE. 2007. pp. 38–47. DOI: 10.1109/ISVD.2007.43.
19. Ho Y.-J., Liu J.-S. Smoothing Voronoi-based obstacle-avoiding path by length-minimizing composite Bezier curve // Proceeding of the International Conference on Service and Interactive Robotics (SIRCon09). Institute of Information Science: Academia Sinica. 2009.
20. Papadopoulou E., Lee D.-T. Critical area computation via Voronoi Diagrams // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1999. vol. 18. no. 4. pp. 463–474. DOI: 10.1109/43.752929.

Зинов Владислав Игоревич — аспирант, кафедра технической кибернетики, Федеральное государственное бюджетное образовательное учреждение высшего образования «Уфимский университет науки и технологий». Область научных интересов: оптимизация в сфере проектирования зданий, эвристические и метаэвристические методы, системы поддержки принятия решений. Число научных публикаций — 8. zinovvladislavufa@gmail.com; улица Заки Валиди, 32, 450076, Уфа, Россия; р.т.: +7(917)435-5560.

УДК 004.021:004.42 **DOI** 10.15622/ia.25.2.10

Е.С. Трушкин, В.И. Фрейман

ПРЕДСКАЗАТЕЛЬНЫЙ МЕТОД РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ НА ОСНОВЕ МНОГОКРИТЕРИАЛЬНОЙ МОДЕЛИ ПРИНЯТИЯ РЕШЕНИЙ

Трушкин Е.С., Фрейман В.И. **Предсказательный метод распределения ресурсов в вычислительных системах на основе многокритериальной модели принятия решений.**

Аннотация. Современные вычислительные системы функционируют, как правило, в условиях гетерогенности и переменной нагрузки. Важным инструментом обеспечения высоких показателей функционирования (например, производительность, надежность, устойчивость) является эффективное распределение вычислительных ресурсов. В связи с этим актуальной проблемой является разработка методов распределения задач, позволяющих улучшать несколько показателей одновременно. Предлагаемый в работе подход представляет собой расширение предсказательного метода распределения ресурсов. Это осуществляется за счет введения многокритериальной модели принятия решений, включающей прогнозное время выполнения, текущую загрузку узлов и достоверность прогноза, оцениваемую по статистике расхождений фактических и прогнозных значений. Объект исследования – вычислительные системы с неоднородными узлами, обрабатывающими потоки задач переменной сложности. Предмет исследования – модели и алгоритмы предсказательного распределения вычислительных ресурсов на основе многокритериального принятия решений. Цель исследования – повышение эффективности и устойчивости функционирования гетерогенных вычислительных систем за счет использования более обоснованного механизма выбора узла на основе совокупности критериев. Выполнен обзор существующих динамических и предсказательных методов распределения, выявлены их преимущества, недостатки и ограничения по эффективному применению. Разработана многокритериальная модель принятия решений, реализующая построение множества Парето-оптимальных решений и процедуру арбитража. Проведено программное моделирование в различных сценариях функционирования системы, включая условия со сниженной достоверностью статистики. Результаты исследования показали, что предлагаемый предсказательный метод на основе многокритериального принятия решений обеспечивает снижение среднего времени выполнения задач и повышение равномерности нагрузки узлов по сравнению с известными подходами. Полученные результаты предлагается использовать при построении гетерогенных вычислительных систем с адаптивными системами управления ресурсами.

Ключевые слова: вычислительная система, методы распределения ресурсов, Парето-оптимальность, многокритериальный выбор, прогнозирование нагрузки, статистические данные.

1. Введение. Одной из ключевых задач, стоящих перед разработчиками современных вычислительных систем, является эффективное распределение вычислительных ресурсов. От того, насколько рационально назначаются задачи на элементы (узлы) вычислительной системы, напрямую зависят ее основные показатели, например, такие как производительность, масштабируемость

и отказоустойчивость. В условиях увеличивающейся динамики потоков задач, разнообразия типов нагрузок и неоднородности узлов становится очевидным, что существующие методы, которые можно разделить на статические и динамические, требуют серьезной модернизации [1].

Статические алгоритмы планирования распределяют задачи по узлам заранее, исходя из известных характеристик задач и ресурсов. Классический пример – методы планирования, ориентированные на графы задач (DAG), такие как алгоритмы HEFT и CPoP [2]. Они рассчитывают приоритеты задач (например, по «весу» выполнения) и закрепляют их за компонентами вычислительных систем (например, процессорами) так, чтобы минимизировать общее время выполнения. Преимущество таких методов – высокая эффективность расписания при выполнении предпосылок (известность времени выполнения и связи задач) и небольшое время планирования [3]. Однако эти алгоритмы сложно адаптировать к изменениям: в статической схеме нет обратной связи с текущей загрузкой, поэтому при колебаниях нагрузки или неточности оценок, такая схема перестает эффективно работать. Как отмечается в работе [4], статические схемы в современных многомерных системах часто оказываются нерезультативными: они плохо справляются с переменным спросом и неоднородностью ресурсов. Так, в опытах по объединенным суперкомпьютерным системам статический подход приводил к высокому уровню сбоев (более 30%), тогда как альтернативное (динамическое) планирование снижало этот показатель до ~8%. Кроме того, задача статического планирования является NP-полной, что вынуждает использовать эвристики: простейшие алгоритмы (например, с поиском лучшего места) при большом числе задач дают высокую вычислительную сложность [5]. Таким образом, ограничения статических методов – это зависимость от корректности исходных данных и отсутствие адаптивности: они не учитывают изменчивость реальных условий и не справляются с неожиданными пиками нагрузки.

Классические динамические методы распределения хорошо изучены и широко применяются благодаря простоте реализации и малым вычислительным издержкам; они основываются на реактивном контроле (мониторинг текущей загрузки, очередей и т.п.) [6 – 7]. Алгоритмы типа Least-Connections [8] легко реализуются и имеют низкие вычислительные затраты, поскольку основываются на простом мониторинге показателей (CPU, очередь запросов и т.п.). Такой подход хорошо подходит для адаптивности: планировщик

быстро реагирует на перегрузку того или иного узла и перераспределяет нагрузку, что значительно повышает эффективность по сравнению со статикой [9]. Например, H. Shim в работе [4] проводит сравнение и показывает, что динамический алгоритм обеспечил до 3,5 раз более эффективное использование ресурсов, чем статический.

Тем не менее, у динамических методов есть ограничения. Во-первых, они принимают решения на основе только текущей информации, игнорируя тренды и исторические закономерности. Это означает, что при резких изменениях нагрузки или в условиях гетерогенности ресурсов реактивный планировщик может совершать повторяющиеся или контрпродуктивные действия, что говорит об ухудшении качества принятия решений [10]. Как показали исследования, реактивные схемы без предсказаний не учитывают предыдущие данные, что вызывает задержки и удлинение времени ожидания задач [11]. Во-вторых, даже при небольшой «стоимости» принятия решений динамика может дать неоптимальные результаты: например, если система перестраивается слишком часто, то растут издержки на переключения и коммуникацию. Кроме того, качественное распределение нагрузки лишь по одному критерию (занятость процессора) часто не учитывает других важнейших параметров (энергоэффективность, приоритеты задач, надежность), что ограничивает эффективность подхода в современных облачных средах.

Предсказательные (предиктивные) методы расширяют динамический подход заложенным прогнозом. Они используют исторические данные [12] и машинное обучение [13] для оценки будущей нагрузки или времени выполнения задач, что позволяет планировать распределение ресурсов заранее. С точки зрения принципа, такие методы могут сочетать текущий мониторинг с моделированием «на что можно рассчитывать в ближайшем будущем». Преимущество прогностических схем – возможность оптимизировать решение (например, заранее подготовить узлы к ожидаемому пику нагрузки, ускорить нужные серверы или более равномерно распределить большие задачи). Обзор современных исследований показывает, что ИИ-решения способны повысить адаптивность и энергоэффективность облачных сред, сократить простои и учесть сложные зависимости между задачами [14 – 15]. Несмотря на широкие возможности, предсказательные методы также имеют ряд ограничений. В первую очередь, точность принимаемых решений напрямую зависит от качества прогноза: ошибки в моделях

машинного обучения могут приводить к неверному выбору узла и, как следствие, к ухудшению производительности системы. Во вторую очередь, построение и поддержание таких моделей требует существенных вычислительных ресурсов и больших объемов обучающих данных, что делает применение подхода затруднительным в средах с ограниченными ресурсами. В работе [16] поднят вопрос о ресурсной стоимости процессов распределения вычислительных ресурсов в динамических, ресурсно-ограниченных вычислительных средах. Проблема рассматривается в плане снижения ресурсной стоимости многокритериальной оптимизации распределения вычислительных ресурсов, однако предложено использование комплексов метаэвристик, с их последовательным выбором и применением в зависимости от ограничений на время получения результата. Так же, ИИ-модели чувствительны к изменению характера нагрузки: при сдвиге распределений или появлении новых типов задач требуется дополнительное (fine-tuning) или новое обучение, иначе прогнозы теряют актуальность и обоснованность [17]. В статье [18] производится сравнение ресурсной эффективности эволюционных алгоритмов с интеграцией ламарковской и болдуинской эволюций в задачах распределения вычислительных ресурсов.

Таким образом, ни один из рассмотренных подходов не лишен недостатков. Статические методы не учитывают изменчивость нагрузки и зависят от точности исходных оценок. Динамические решения просты и быстры, но ориентированы лишь на текущие показатели, не предвидят изменения условий и могут сбалансировать систему всего по одному параметру. Прогностические схемы более гибкие, однако требуют сложной инфраструктуры ИИ: большие вычислительные затраты и массив обучающих данных в совокупности с чувствительностью к ошибкам прогнозов ограничивают их надежность. Такая совокупность проблем обуславливает потребность в более устойчивых методах, способных учитывать сразу несколько факторов при выборе назначения задачи.

Авторами в работе [19] был предложен предсказательный метод распределения, который применим в системах с жесткими ограничениями по вычислительным ресурсам. Он отличается от аналогичных тем, что прогнозирование будущей нагрузки и адаптивное перераспределение вычислительных задач выполняется с учетом совокупности текущего состояния системы и статистической информации, о предыдущих схожих по характеристикам задачах. Такой подход позволяет заранее оценивать будущую загруженность вычислительных узлов и принимать более обоснованные решения

о распределении ресурсов. Однако ранее авторами не рассматривался вопрос ошибок в прогнозах. Поэтому возникает необходимость в расширении модели принятия решений, реализуемой в рамках предсказательного метода, до многокритериальной модели, способной учитывать несколько факторов одновременно. Настоящая статья развивает этот подход и формализует процесс выбора узла, как многокритериальную задачу. Ключевым отличием является введение критерия достоверности прогноза, построенного на статистике расхождений между прогнозным и фактическим временем выполнения.

Под вычислительной задачей в настоящей работе понимается законченный процесс или работа, требующая выделения ресурсов узла (CPU, памяти, дисковых операций и т.д.) и обладающая определенными характеристиками (например, оценочным временем выполнения или приоритетом). Такие задачи в реальных системах могут существенно различаться по своей природе, требованиям и временным ограничениям. Приведем некоторые типичные примеры практических задач, встречающихся в современных вычислительных средах:

1. Научные и инженерные вычисления. К ним относятся ресурсоемкие пакетные задачи на суперкомпьютерах и кластерах. Примеры – численное моделирование физических процессов (гидродинамика, аэродинамика), климатические расчеты, обработка последовательностей ДНК, визуализация научных данных, финансовое и химико-биологическое моделирование и др. Такие задачи часто запускаются как batch processing, которые пользователи отправляют на планировщик кластера. Планировщик распределяет эти задания по свободным узлам кластера для параллельного выполнения. Данные приложения требуют большего вычислительного времени и, как правило, имеют заранее известную структуру задачи (например, граф зависимостей) [20].

2. Веб-сервисы и микросервисы. Каждый HTTP-запрос или транзакция к веб-приложению интерпретируется системой как отдельная задача (процесс или поток), которую нужно быстро обработать. Примеры – это генерация веб-страниц, обращение к API, выполнение запросов к базам данных и т.п. Такие задачи обычно недолговечны (от нескольких миллисекунд до секунд) и имеют строгие требования по задержке (латентности). Планировщики распределяют их по серверам и виртуальным машинам в реальном времени, чтобы обеспечить высокую пропускную способность и быстрый отклик сервисов. В отличие от batch processing, здесь

оптимизация идет по минимизации задержек и соблюдению приоритетов каждого запроса [21].

Цель данной работы – улучшить предложенный ранее предсказательный метод распределения задач за счет многокритериального принятия решения при выборе узла.

Задачи статьи:

1. Формализовать многокритериальную модель принятия решений, определить показатели эффективности работы вычислительной системы (раздел 2).
2. Ввести в перечень ключевых критериев, кроме традиционно используемых (прогнозируемое время выполнения, текущее/ожидаемое значение загрузки узла), новый – метрику надежности/достоверности прогноза (раздел 3).
3. Применить подход для оптимального назначения задачи, сочетающий построение Парето-множества кандидатов и последующую процедуру арбитража/ранжирования с учетом заданных предпочтений (разделы 4 – 5).
4. Разработать алгоритм многокритериального принятия решений как основного этапа предсказательного метода распределения ресурсов в вычислительных системах (раздел 6).
5. Создать программу имитационного моделирования и провести эксперименты для сравнения предсказательного метода многокритериального принятия решений с предсказательным однокритериальным и динамическим (раздел 7 – 8).

2. Постановка задачи. В рамках рассматриваемой работы проблема распределения ресурсов формулируется как назначение вычислительной задачи на один из доступных узлов в условиях многокритериального выбора с использованием статистической информации.

Рассматривается система, состоящая из множества вычислительных узлов:

$$U = \{u_1, u_2, \dots, u_n\}.$$

Каждая вновь поступающая задача Z должна быть назначена на один из доступных узлов до начала ее выполнения.

Условия функционирования системы:

− узлы являются гетерогенными и различаются по производительности,

– система работает в условиях переменной нагрузки: количество поступающих задач и их характеристики (например, требуемая вычислительная сложность) со временем изменяются,

– в случае отсутствия свободных узлов задача будет поставлена в очередь или получит отказ в обслуживании при заполнении очереди.

Показателями, по которым оценивается эффективность системы, в данной работе выбраны быстродействие (время обработки задач) и равномерность распределения нагрузки между узлами системы.

Постановка задачи сводится к многокритериальной оптимизации: необходимо выбрать узел $u^* \in U$, для которого совокупность выбранных критериев будет оптимальной с учетом ограничений.

Пусть назначение задачи Z на узел u_j описывается кортежем критериев:

$$C(u_j, Z) = \langle c_1(u_j, Z), c_2(u_j, Z), \dots, c_k(u_j, Z), \dots, c_n(u_j, Z) \rangle,$$

где c_k – критерий качества, $k \in [1, n]$ (например: время выполнения, загрузка, вероятность отказа, достоверность прогноза и др.); n – количество критериев. Тогда задача распределения ресурсов формулируется как выбор узла, для которого кортеж $C(u_j, Z)$ будет иметь наилучшую совокупность критериев. Для решения поставленной задачи далее будут предложены критерии и подходы для нахождения их оптимальной совокупности.

3. Критерии оценки узлов. В общем виде, число критериев качества в задаче распределения ресурсов не фиксировано и может быть неограниченно большим. Увеличение числа критериев n влияет на несколько важных аспектов [22]:

– расширяются пространство компромиссов между критериями,

– растет объем данных для сравнения кандидатов (узлов),

– повышается вычислительная сложность процедур фильтрации и последующего ранжирования,

– снижается наглядность интерпретации результатов. Поэтому при практической реализации обычно выбирают ограниченное подмножество информативных и интерпретируемых показателей (критериев) – применяют методы отбора/редукции признаков [23], кластеризацию в пространстве критериев [24] или скаляризацию [25].

В данной работе, как наиболее типичные для распределенных вычислительных систем, используются три критерия, отражающих функционирование узлов:

1. Прогнозное время выполнения задачи – ожидаемая длительность выполнения задачи Z на узле u_j, рассчитанная по модели прогнозирования:

$$c_1 = T_{\text{п}}(u_j, Z).$$

Прогнозируемое значение параметра определяется в зависимости от расчетного значения и реальных (статистических) значений обработки аналогичного:

$$T_{\text{п}}(r) = \alpha T_{\text{р}}(r) + \sum_{s=1}^{N} \beta_s T_{\text{с}}(r-s),$$

где $T_{\text{п}}(r)$ – прогнозируемое значение на r-м шаге; $T_{\text{р}}(r)$ – расчетное значение на r-м шаге; $T_{\text{с}}(r–1)$ … $T_{\text{с}}(r–N)$ – статистические значения предыдущей обработки; N – глубина памяти (или максимальное значение количества имеющихся данных); α – весовой коэффициент расчетного значения, β_s – весовой коэффициент s-го статистического значения.

Этот критерий отражает быстродействие узла.

2. Текущая загрузка узла – время, которое необходимо узлу, чтобы обработать уже назначенные ему задачи.

$$c_2 = T_{\text{п}}(u_j, Z_{i-1}) + T_{\text{п}}(u_j, Z_{i-2}) + \cdots + T_{\text{п}}(u_j, Z_{i-m}),$$

где m – количество предыдущих задач.

Учет этого критерия позволяет распределять задачи равномерно и предотвращать перегрузку отдельных узлов.

3. Достоверность прогноза – показатель, отражающий статистические погрешности прогнозирования на конкретном узле.

Рассчитывается на основе двух рядов наблюдений:

- прогнозное время выполнения задачи: $T_{\text{п},i}(u_j, Z)$,
- фактическое время выполнения задачи: $T_{\text{ф},i}(u_j, Z)$.

Погрешность прогноза для каждого момента времени (шага) i определяется следующим образом:

$$\Delta T_i(u_j, Z) = T_{\text{ф},i}(u_j, Z) - T_{\text{п},i}(u_j, Z).$$

Для дальнейших расчетов предлагается использовать также значение абсолютной погрешности:

$$\Delta A_i = |\Delta T_i|.$$

Далее определяются необходимые характеристики погрешности. Для качественной оценки прогноза предлагается использовать характеристики погрешности на скользящем окне M. Это окно ограничено условием $M \leq i$, где i – число накопленных наблюдений, если статистики недостаточно ($i < M$), то характеристики рассчитываются по доступным данным ($M = i$).

Среднее значение погрешности прогнозирования:

$$\Delta T_{\text{ср}}(u_j, Z) = \frac{1}{M} \sum_{m=i-M}^{i} \Delta T_m(u_j, Z), i \geq M,$$

$$\Delta T_{\text{ср}}(u_j, Z) = \frac{1}{i} \sum_{m=0}^{i} \Delta T_m(u_j, Z), M > i \geq 0.$$

В случае, когда $\Delta T_{\text{ср}}(u_j, Z) > 0$ это означает, что на узле фактическое время достаточно часто больше прогнозного. А это значит – есть риск срыва сроков выполнения задач, которые важно обработать вовремя. Если $\Delta T_{\text{ср}}(u_j, Z) < 0$, то узел систематически обрабатывает задачу быстрее прогнозного времени – возможно, часть ресурсов не используется.

Стоит отметить, что возможна ситуация, когда $\Delta T_{\text{ср}}(u_j, Z) = 0$. В этом случае могли возникнуть следующие проблемы:

1. Наблюдаемый узел имеет одинаковую среднюю погрешность, как в большую сторону, так и в меньшую (например, {+2, -2, +2, -2}).
2. На узле наблюдалась существенная погрешность (+ или -), которая компенсировала величину суммы противоположных погрешностей (например, {+8, -2, -2, -2, -2}.

Для решения этих проблем используем расчет среднего значение абсолютной погрешности прогнозирования:

$$\Delta A_{\text{ср}}(u_j, Z) = \frac{1}{M} \sum_{m=i-M}^{i} \Delta A_m(u_j, Z), i \geq M,$$

$$\Delta A_{\text{ср}}(u_j, Z) = \frac{1}{i}\sum_{m=0}^{i} \Delta A_m(u_j, Z), M > i \geq 0.$$

$\Delta A_{\text{ср}}$ показывает на сколько в среднем прогнозируемое значение отличается от фактического. В отличие от простого среднего значения погрешности, абсолютное не обнуляется при чередовании (+) и (−). Чем меньше $\Delta A_{\text{ср}}$, тем точнее прогноз в среднем.

Также, кроме описанных выше проблем, может возникнуть неопределенность, при которой на нескольких узлах абсолютная погрешность $\Delta A_{\text{ср}}$ будет одинакова. Для того чтобы избежать этого, предлагается использовать расчет среднеквадратического отклонения погрешностей σ:

$$c_3 = \sigma(u_j, Z) = \sqrt{\frac{1}{M-1}\sum_{m=i-M}^{i} \left(\Delta T_m(u_j, Z) - \Delta T_{\text{ср}}(u_j, Z)\right)^2}.$$

Этот расчет показывает, насколько отклоняется погрешность прогноза от среднего значения. Если σ достаточно велико, значит за последние M отсчетов фактическое значение сильно отличалось от прогнозного.

Далее рассмотрим числовой пример. Предположим, что имеется два узла (А и Б), среди которых необходимо выбрать один для обработки задачи.

1. Узел А: погрешности [+2, -2, +2, -2].

$$\Delta A_{\text{ср}} = 2$$
$$\Delta T_{\text{ср}} = 0$$
$$\sigma \approx 2{,}31$$

2. Узел Б: погрешности [+4, -2, -2, 0].

$$\Delta A_{\text{ср}} = 2$$
$$\Delta T_{\text{ср}} = 0$$
$$\sigma \approx 2{,}83$$

Оба узла имеют одинаковый $\Delta A_{\text{ср}} = 2$ и $\Delta T_{\text{ср}} = 0$, но среднеквадратическое отклонение у них разное. Это позволяет сделать

вывод: узел А: $\sigma \approx 2{,}31$ – прогноз более точен (разброс погрешности в меньшем диапазоне). Узел В: $\sigma \approx 2{,}83$ – прогноз менее точен (разброс погрешности в большем диапазоне).

Использование выбранных трех критериев обусловлено их практической релевантностью для задач распределения вычислительных задач: c_1 отражает производительность, c_2 – состояние загрузки, c_3 – точность прогнозов и устойчивость работы системы при неопределенности. В каждом конкретном случае перечень критериев может быть изменен – соответствующие последствия для методов отбора и вычислительной сложности обсуждаются в разделах 4 и 5, где также приводится используемый алгоритм предварительной фильтрации узлов и последующего выбора.

4. Концепция Парето-оптимальности. В многокритериальной постановке, как правило, сложно выделить один узел, который был бы наилучшим по всем показателям. Для этого используется понятие Парето-доминирования [26].

Парето-доминирование:

Пусть два узла u_i и u_j имеют соответствующие кортежи критериев $C(u_i)$ и $C(u_j)$.

Тогда u_i доминирует u_j, если:

- для всех $k \in [1, n]$: $c_k(u_i) \leq c_k(u_j)$,
- существует хотя бы один k': $c_{k'}(u_i) < c_{k'}(u_j)$.

Парето-множество:

Множество узлов, которые не доминируются ни одним другим, образует Парето-фронт или множество Парето-оптимальных решений:

$$P = \{ u_j \in U \mid \nexists u_i \in U : C(u_i) \prec C(u_j) \}.$$

Множество P содержит все такие элементы $u_j \in U$, для которых не существует другого элемента $u_i \in U$, доминирующего u_j. Парето-множество отражает набор компромиссных решений. Каждый элемент этого множества является обоснованным с какой-либо точки зрения, и не существует универсального предпочтения без дополнительной информации. Выбор из этого множества требует сужения на основе предпочтений системы или пользователя [27].

Построение Парето-фронта связано с процедурой перебора всех критериев для каждого узла. Поэтому вычислительная сложность формирования Парето-фронта зависит от количества узлов и количества критериев. Характер зависимостей и ограничения применения данного метода являются темой дальнейших исследований.

5. Роль арбитража и лица, принимающего решение (ЛПР). Построение множества Парето-оптимальных решений позволяет выделить и отбросить те варианты распределения задачи (или перечня задач), которые являются неудовлетворительными по сравнению с остальными. Однако отсутствие доминирования не означает равную предпочтительность – необходимо выбрать конкретный узел для назначения задачи. Этот выбор невозможно осуществить без учета предпочтений некоторого субъекта – лица, принимающего решение (ЛПР) [28]. В случае автоматического управления распределением задач, что и имеет место в вычислительных системах, эту функцию выполняет программный модуль принятия решения (планировщик).

Для того, чтобы принять решение, применяются методы, которые описаны ниже. Они не гарантируют, что останется одно из возможных решений, но позволяют существенно уменьшить их количество. После чего выбирается любой из оставшихся вариантов, так как они считаются равнозначными.

Метод агрегирования (линейная свертка).

При наличии кортежа весов $w = \langle w_1, w_2 \dots, w_m \rangle$, где $w_k \geq 0$ и $\sum w_k = 1$, вводится функция полезности:

$$S(u_j, Z) = \sum_{k=1}^{m} w_k \cdot c_k(u_j, Z).$$

Каждому критерию c_k сопоставляется коэффициент значимости w_k. После чего выбирается узел с минимальным (или максимальным – в зависимости от задачи) значением функции полезности. Преимущества: простота, гибкость. Недостатки: необходимость задания весов, возможность потери информации о компромиссах [29].

Иерархическая фильтрация.

В данном методе критерии ранжируются по приоритету. Сначала фильтруются узлы по главному критерию (например, прогнозируемое время обработки), затем – по следующему и т.д., пока не останется один или несколько вариантов. Метод подходит в случаях, когда некоторые критерии являются жесткими ограничениями, например, не назначать задачи на узел с высокой погрешностью прогнозирования. Преимущества: высокая управляемость, логичность. Недостатки: может отбрасывать близкие по значению, но в целом более выгодные решения [30].

Также существуют методы и модели для принятия решений, которые включают более сложные механизмы выбора. Примеры: логические правила (если… то…) [31], лексикографический порядок, нейросетевые функции предпочтений [32].

Интеграция механизма арбитража и моделирования ЛПР позволяет системе переходить от «чистой» оптимизации к интеллектуальному принятию решений, учитывающему контекст, цели, политику управления и пользовательские требования. Это является ключевым шагом на пути к созданию адаптивного планировщика, способного устойчиво функционировать в широком диапазоне условий

6. Алгоритм реализации многокритериального предсказательного метода распределения задач. Предложенная в разделах 2 – 5 математическая модель реализуется в виде алгоритма, который активируется при поступлении каждой новой задачи в систему. Алгоритм работает в реальном времени и состоит из следующих последовательных шагов.

Шаг 1. Инициализация и сбор актуального состояния системы.

Перед обработкой новой задачи система выполняет сбор информации о текущем состоянии вычислительных узлов $U = \{u_1, u_2, \ldots, u_n\}$. Формируются показатели:

- текущая загрузка каждого узла (количество активных задач, оставшееся время их выполнения, использование ресурсов),
- обновленная статистика по ранее выполненным задачам (прогнозные и фактические времена),
- доступность узла для приема новой задачи.

Эта информация формирует исходные данные для прогноза и оценки критериев.

Шаг 2. Прогнозирование времени выполнения новой задачи.

Для каждой пары «задача Z – узел u_j» рассчитывается прогнозное время $T_{\text{п},i}(u_j, Z)$ на основании модели прогнозирования, использующей накопленные статистические данные и параметры задачи (например: объем вычислений, тип задачи, требуемые ресурсы). Если статистика по данному типу задач или узлу ограничена, применяется адаптивная аппроксимация на основе близких классов задач. Результаты прогноза сохраняются в таблицу критериев.

Шаг 3. Формирование кортежа критериев и условий доминирования.

Для каждого узла формируется кортеж критериев:

$$C(u_j, Z) = \langle c_1(u_j, Z), c_2(u_j, Z), \ldots, c_k(u_j, Z), \ldots, c_n(u_j, Z) \rangle.$$

После чего эти кортежи сводятся в таблицу значений критериев для всех рассматриваемых узлов (таблица 1).

Таблица 1. Результирующая таблица критериев

	Критерии					
Номер узла	c_1	c_2	…	c_i	…	c_n
1	$c_1(u_1, Z)$	$c_2(u_1, Z)$	…	$c_k(u_1, Z)$	…	$c_n(u_1, Z)$
2	$c_1(u_2, Z)$	$c_2(u_2, Z)$	…	$c_k(u_2, Z)$	…	$c_n(u_2, Z)$
…	…	…	…	…	…	…
j	$c_1(u_j, Z)$	$c_2(u_j, Z)$	…	$c_k(u_j, Z)$	…	$c_n(u_j, Z)$
…	…	…	…	…	…	…
m	$c_1(u_m, Z)$	$c_2(u_m, Z)$	…	$c_k(u_m, Z)$	…	$c_n(u_m, Z)$

Ключевое требование этого шага – однозначно задать отношение «лучше/хуже» для каждой компонентной величины. Для каждого критерия $k = [1, n]$ фиксируется правило:

- либо «меньше – лучше» (минимизационный критерий),
- либо «больше – лучше» (максимизационный критерий).

На основе этих компонентных отношений задается отношение доминирования между двумя узлами. Эти данные служат входом для шага 4 – фильтрации (выделение недоминируемых решений) и последующей сортировки/отбора.

Шаг 4. Построение множества Парето-оптимальных решений.

После определения критериев и построения кортежей производится их анализ для всех узлов. Для каждой пары (u_i, u_j) выполняется проверка на Парето-доминирование. В множество P включаются только те узлы, которые недоминируются никаким другим.

Таким образом, множество P определяет набор альтернатив, являющихся компромиссно оптимальными по совокупности критериев. Построение множества Парето-оптимальных решений реализовано методом полного перебора, что является эффективным при небольшом числе узлов и обеспечивает точное выделение недоминируемых решений.

Шаг 5. Принятие окончательного решения (Арбитраж).

Если множество P содержит более одного узла, выполняется процедура арбитража, реализующая предпочтения лица, принимающего решение (ЛПР), или заданную политику планировщика (возможные стратегии были описаны в разделе 5). Результатом арбитража является окончательное решение о назначении задачи Z на узел $u^* \in P$.

Шаг 6. Выполнение задачи и обновление статистики.

После назначения задачи происходит ее выполнение на выбранном узле u^*. По завершении фиксируются фактические показатели: время выполнения, использованные ресурсы, возникшие отклонения от прогноза. Эти данные добавляются в статистическую базу и используются для пересчета параметров прогнозной модели. Таким образом, система обладает свойством самообучения и повышает точность прогнозов при длительной эксплуатации.

7. Программное моделирование. Для экспериментальных исследований авторами ранее использовался AnyLogic [33]. Для более гибкой настройки разработанная многокритериальная модель была реализована в виде программного комплекса на языке Python, включающего графический интерфейс пользователя и средства визуального анализа результатов. Реализация выполнена в модульной архитектуре и позволяет проводить серию вычислительных экспериментов для сравнения различных методов распределения задач в вычислительных системах.

Эксперименты направлены на сравнительный анализ трех методов распределения:

1. Динамического (Д) – это существующий метод, в котором задачи назначаются на узел с минимальной текущей загрузкой.
2. Предсказательного однокритериального (П) – это авторский метод, согласно которому выбор узла с минимальным прогнозным временем выполнения.
3. Многокритериального предсказательного (М) – этот метод является улучшенной версией предсказательного однокритериального метода (П), в нем выбор узла происходит на основе совокупности критериев (например, прогнозного времени, загрузки узла, достоверность прогноза).

Серия экспериментов выполнена поэтапно, с постепенным усложнением исходных условий, что позволяет проследить изменение поведения алгоритмов при переходе от идеальных к реальным сценариям.

Для моделирования рассматривалась система, имеющая следующие входные данные:

- количество задач,
- группы сложности задач,
- вычислительная сложность каждой группы задач, в условных единицах (у.е.),
- пропорция распределения задач по группам сложности,
- производительность, в у.е./е.в. (единица времени).

Все входные данные являются синтетическими.

7.1. Сценарий 1. Идеальные условия – равномерные и стабильные узлы. Цель: проверить корректность работы модели в условиях, когда отсутствуют различия между узлами и статистика прогноза полностью совпадает с фактическими значениями. В таких условиях любой рациональный метод назначения задач должен приводить к одинаковым результатам по времени выполнения и распределению нагрузки.

Условия эксперимента:

- все узлы имеют одинаковую производительность,
- задачи однородные, с одинаковой вычислительной сложностью,
- исторические данные не содержат выбросов, то есть статистика стабильна,
- количество задач и длина временного ряда выбраны так, чтобы прогноз полностью совпадал с фактическим временем обработки задач.

Ожидаемые результаты:

- методы Д, П и М обеспечивают одинаковое время выполнения и равномерную загрузку узлов,
- Парето-множество в методе М будет содержать все узлы, так как критерии идентичны,
- достоверность прогноза максимальна, а арбитраж не влияет на результат.

На рисунке 1 продемонстрирована конфигурация первого сценария.

В этих условиях все три метода – Д, П, М должны демонстрировать одинаковое поведение. Результаты моделирования этого сценария показаны на рисунке 2.

Рисунок 2 иллюстрирует результаты экспериментов для сценария, в котором нагрузка распределяется равномерно и отсутствуют существенные колебания входных параметров. Как видно из таблицы, при увеличении количества задач от 1000 до 5000 все методы: динамический, однокритериальный предсказательный и многокритериальный предсказательный – демонстрируют одинаковые значения как расчетного, так и фактического времени обработки. Это отражено в одинаковых значениях столбцов «Динамический (расчетное)», «Динамический (фактическое)», «Предсказательно однокритериальный (р/ф)» и «Предск. многокритериальный (р/ф)», где наблюдаются идентичные величины (300, 600, 900, 1200, 1500 мс). Кроме того, показатели баланса загрузки на узлы во всех случаях равны нулю (показывают дельту загрузки между самым нагруженным узлом и самым ненагруженным), что подтверждает отсутствие различий между методами.

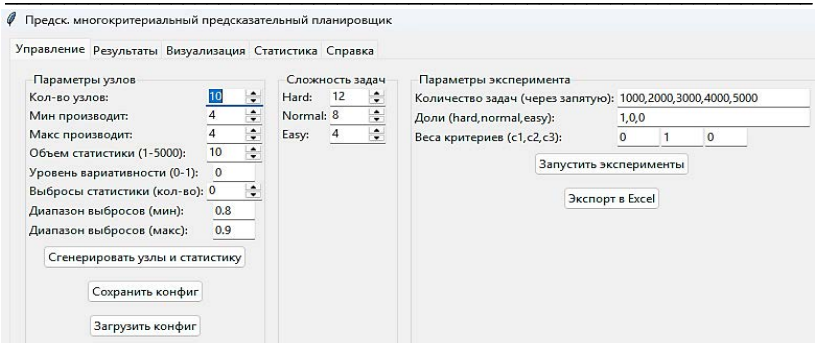
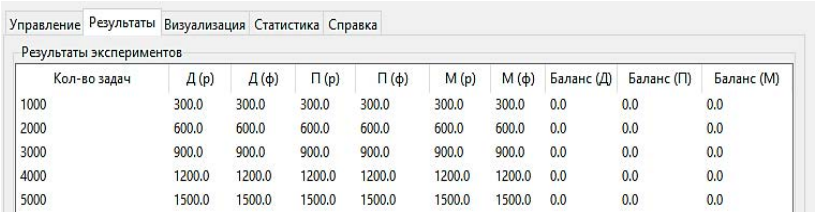


Рис. 1. Окно конфигурации сценария 1



Кол-во задач	Д (р)	Д (ф)	П (р)	П (ф)	М (р)	М (ф)	Баланс (Д)	Баланс (П)	Баланс (М)
1000	300.0	300.0	300.0	300.0	300.0	300.0	0.0	0.0	0.0
2000	600.0	600.0	600.0	600.0	600.0	600.0	0.0	0.0	0.0
3000	900.0	900.0	900.0	900.0	900.0	900.0	0.0	0.0	0.0
4000	1200.0	1200.0	1200.0	1200.0	1200.0	1200.0	0.0	0.0	0.0
5000	1500.0	1500.0	1500.0	1500.0	1500.0	1500.0	0.0	0.0	0.0

Рис. 2. Окно результатов сценария 1

Такой результат объясняется тем, что в данном сценарии выбор узла не влияет на итоговое время выполнения: все узлы находятся в одинаковых условиях, и даже сложные методы (П или М) не получают преимущества. Таким образом, рисунок 2 демонстрирует ситуацию, в которой эффективность всех подходов выравнивается из-за идеальной однородности вычислительной среды и отсутствии погрешности в статистической выборке.

Интерпретация: данный сценарий служит базовой точкой проверки корректности модели – полученные результаты подтверждают корректность реализации алгоритма и отсутствие искусственных искажений: модель в идеальных условиях не дает преимуществ ни одному из методов, что соответствует теоретическим ожиданиям.

7.2. Сценарий 2. Неоднородные узлы, разнотипные задачи и статистические данные – преимущество предсказательных методов. Цель: проверить корректность работы модели и показать, что при различной производительности узлов и сложности задач, а также при добавлении статистических данных предсказательные методы обеспечивают более рациональное распределение по сравнению с динамическим распределением. Это объясняется тем, что введение

статистики позволяет оценивать не только расчетные показатели производительности, но и учитывать их фактические значения.

Для доказательства адекватности разработанной модели многокритериальный метод принятия решения (Парето-оптимизация на первом этапе и линейная свертка на втором) используется в однокритериальном режиме.

Условия эксперимента:

- узлы имеют различную производительность, что отражает гетерогенность вычислительных ресурсов (например, различие в тактовой частоте),
- в поток задач включаются три класса по сложности: простые задачи низкой трудоемкости, задачи средней сложности, ресурсоемкие задачи, требующие максимальной производительности,
- соотношение типов задач предлагается принять как соотношение 34 % : 33 % : 33 % (один из возможных вариантов),
- для каждой пары (тип задачи, узел) накоплена статистика выполнения, позволяющая построить корректный прогноз времени выполнения для приближения к соответствующим фактическим (реальным) значениям,
- система не моделирует текущую загрузку – предполагается, что все узлы доступны и готовы к приему новой задачи (равные начальные условия),
- весовые коэффициенты многокритериального метода заданы так, чтобы принятие решения осуществлялось по одному критерию, как и в однокритериальном методе.

Ожидаемые результаты:

- динамический метод начнет показывать худший результат, так как ориентируется только на текущую загрузку и фиксированные расчетные значения производительности, игнорируя изменчивость реальной производительности устройств,
- однокритериальный предсказательный метод и многокритериальный покажут близкие результаты по времени выполнения и балансировке, так как веса критериев метода (М) заданы таким, образом, чтобы он соответствовал методу (П).

Конфигурация программы для второго сценария представлена на рисунке 3.

На рисунке 4 представлено сравнение среднего времени выполнения задач для трех рассматриваемых методов распределения: Д, П и М. По оси вертикали откладывается время выполнения задач, где меньшие значения соответствуют более эффективной работе системы.

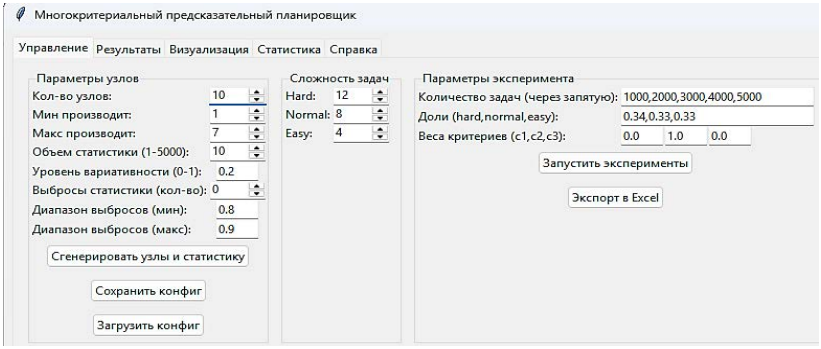


Рис. 3. Окно конфигурации сценария 2

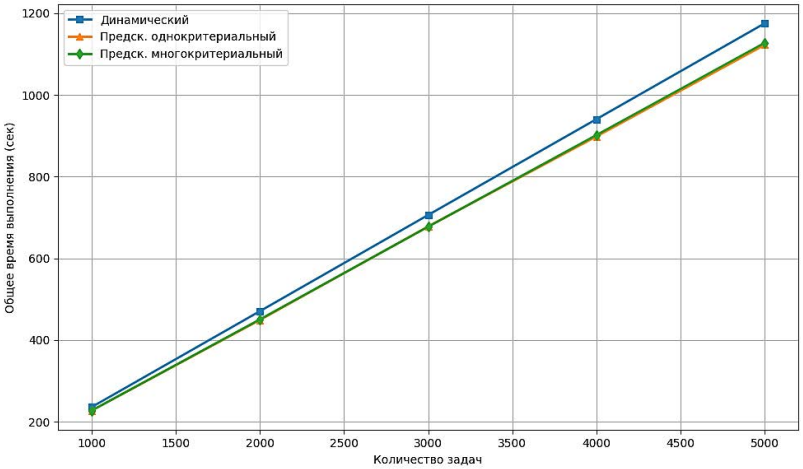


Рис. 4. График сравнения времени выполнения задач для сценария 2 (меньше – лучше)

На рисунке 5 приведено сравнение степени равномерности загрузки вычислительных узлов при использовании трех методов распределения задач. По вертикальной оси отложена разница между максимальной и минимальной загрузкой узлов, характеризующая степень дисбаланса системы: чем меньше значение, тем более равномерно распределена нагрузка.

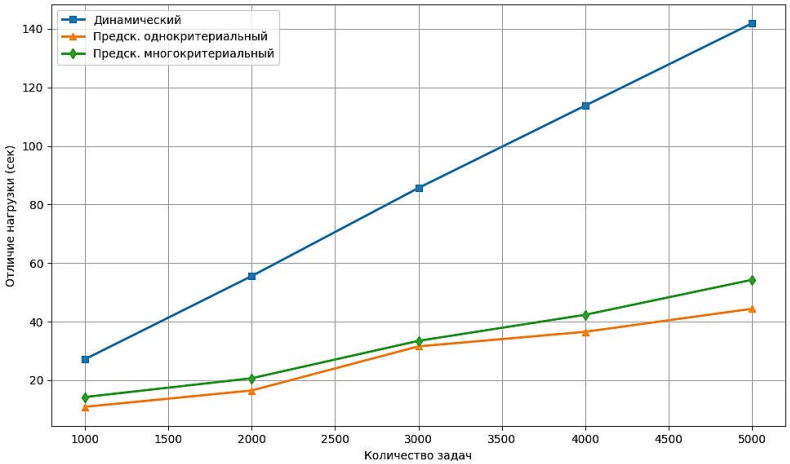


Рис. 5. График сравнения балансировки нагрузки узлов для сценария 2 (меньше – лучше)

Так как на рисунках 4, 5 представлены результаты в абсолютных значениях, для лучшей наглядности предлагается оценивать методы между собой в относительных значениях. Эти значения получаются следующим образом:

$$r_{\text{отн}} = 100 - \frac{r_{\text{абс}}(\text{П}) \cdot 100}{r_{\text{абс}}(\text{Д})},$$

где $r_{\text{отн}}$ – относительное отличие методов (%), $r_{\text{абс}}$ – абсолютное значение показателя (время обработки задач, равномерность загрузки) для рассматриваемых методов.

На рисунках 6, 7 показано относительное отличие предсказательных методов от динамического по двум вышеупомянутым показателям: время выполнения и равномерность.

Из представленных данных видно, что динамический метод демонстрирует наибольшее время выполнения и обеспечивает наихудшую балансировку среди рассматриваемых методов. Это обусловлено тем, что данный метод принимает решение исключительно на основе текущей загрузки узлов и не использует накопленную статистическую информацию о фактической производительности. Предсказательные методы (П и М), наоборот, показывают меньшие значения времени выполнения задач и более сбалансированное распределение.

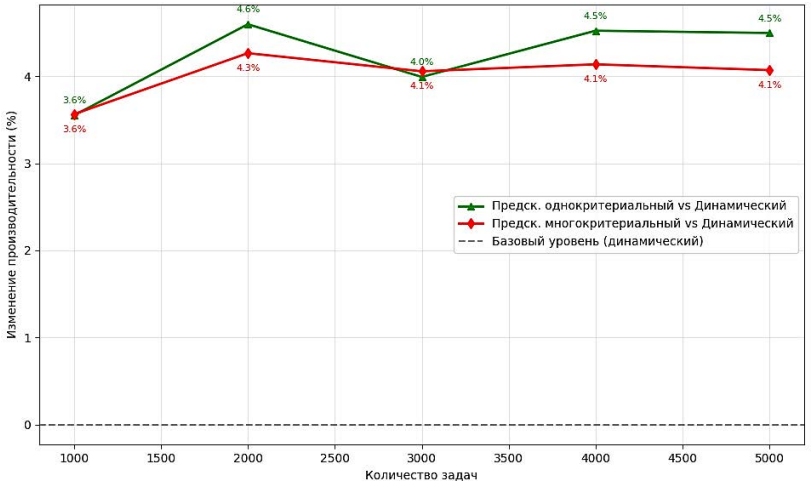


Рис. 6. График относительного изменения времени выполнения для сценария 2 (больше – лучше)

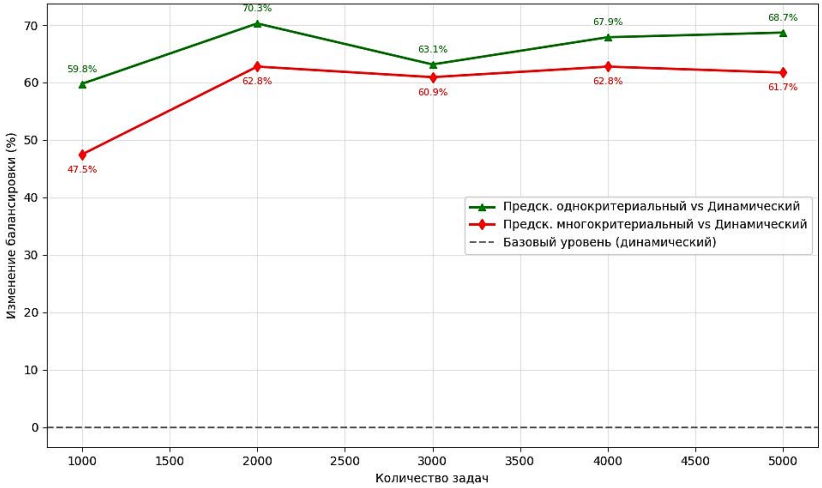


Рис. 7. График относительного изменения балансировки нагрузки для сценария 2 (больше – лучше)

Стоит отметить, что различия между предсказательным однокритериальным и многокритериальным методами в данном сценарии минимальны. Это связано с выбором весовых коэффициентов критериев в функции агрегированной оценки в методе (M). В рассматриваемой конфигурации использовано значение

кортежа весов $w = \langle 0, 1, 0 \rangle$, при котором единственным значимым критерием становится прогнозное время выполнения задачи. В результате оба предсказательных метода (П и М) выбирают одни и те же узлы или узлы близкие по характеристикам для размещения задач и обеспечивают сопоставимый уровень эффективности.

Статистика необходима для оценки реальной производительности, она может показать, что производительность узла изменилась по тем или иным причинам (например, из-за повышения фоновой нагрузки выделенные вычислительные ресурсы (оперативная память, файл подкачки) уменьшились, или из-за повышения температуры могла снизиться тактовая частота процессора).

7.3. Сценарий 3. Нестабильные условия – влияние достоверности прогноза.

Цель: выявить различия между предсказательными методами при ухудшении качества прогноза и наличии выбросов в статистических данных.

Условия эксперимента:

- в статистику выполнения задач добавляются случайные выбросы,
- узлы сохраняют различную производительность,
- для многокритериального метода добавляется критерий достоверности прогноза с весом $c_3 > 0$.

Ожидаемые результаты:

- однокритериальный предсказательный метод (П) начнет назначать задачи на узлы с недостоверной статистикой, что ведет росту общего времени выполнения и может способствовать перегрузке отдельных узлов;
- многокритериальный метод (М) учитывает достоверность прогноза и частично скомпенсирует выбросы, сохранив сбалансированное распределение;
- отличие в эффективности между П и М становится значимой: многокритериальный метод обеспечит наименьшее время выполнения, однако «пожертвует» равномерной загрузкой узлов, что объясняется весовыми коэффициентами.

Конфигурация программы для третьего сценария представлена на рисунке 8.

Ниже представлены результаты сравнения предлагаемого и известных методов для сценария 3 по нескольким критериям - время выполнения задач (рисунок 9), балансировка нагрузки (рисунок 10), относительное изменение времени выполнения задач (рисунок 11), относительное изменение балансировки нагрузки (рисунок 12).

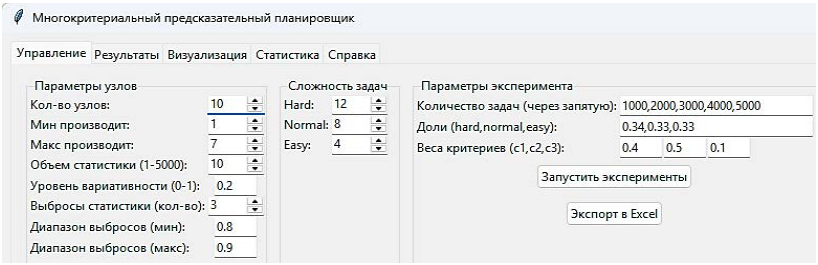


Рис. 8. Окно конфигурации сценария 3

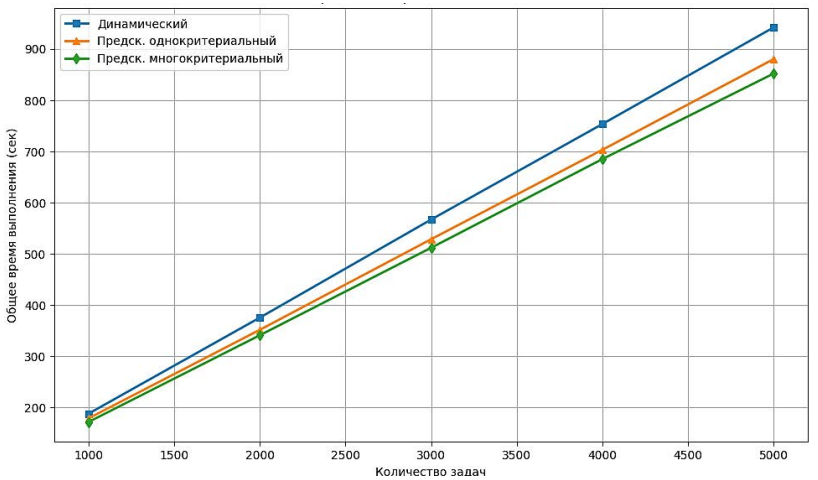


Рис. 9. График сравнения времени выполнения задач для сценария 3 (меньше – лучше)

Переход от однокритериального к многокритериальному предсказательному подходу обусловлен необходимостью учитывать не только текущую загруженность узла, но и контекстные характеристики системы: прогнозное время выполнения и достоверность прогноза, или др. параметры (например, энергопотребление узла, приоритет задач и т. д).

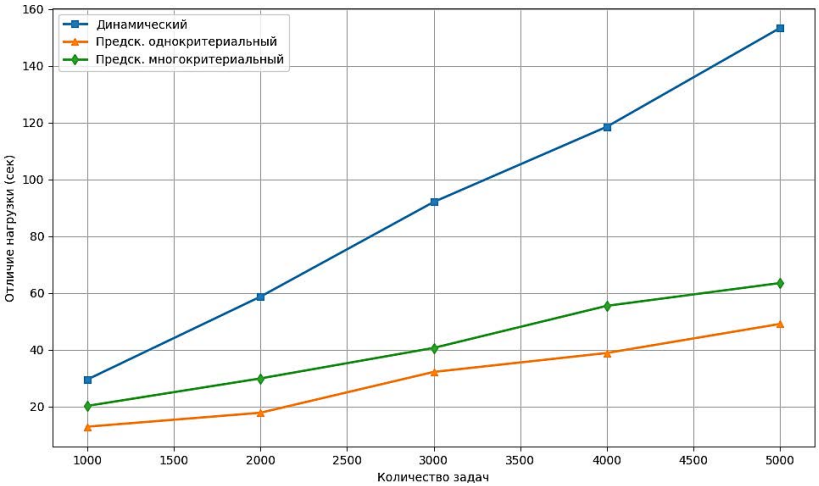


Рис. 10. График сравнения балансировки нагрузки узлов для сценария 3 (меньше – лучше)

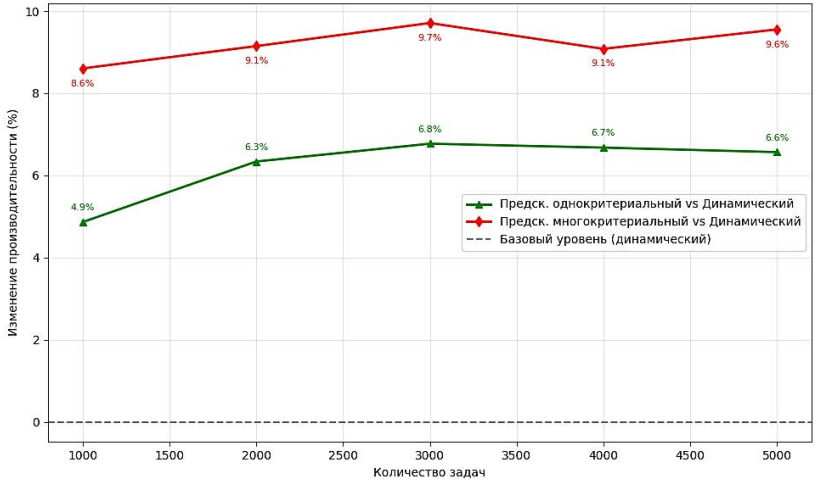


Рис. 11. График относительного изменения времени выполнения для сценария 3 (больше – лучше)

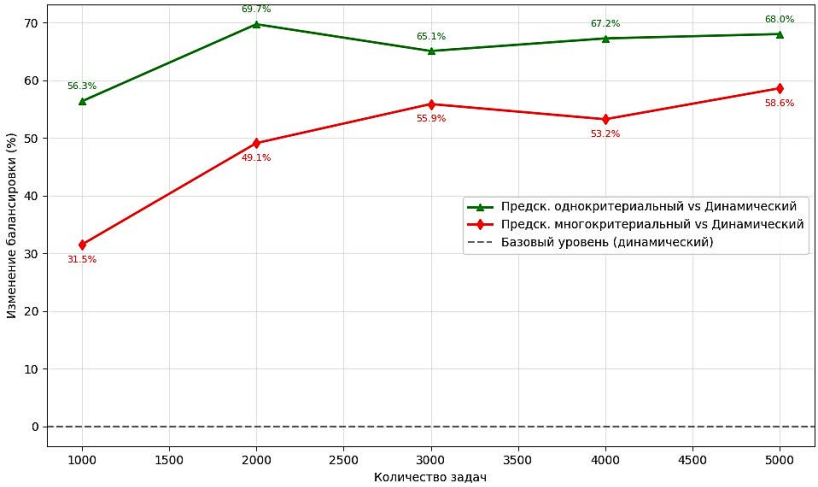


Рис. 12. График относительного изменения балансировки нагрузки для сценария 3 (больше – лучше)

8. Анализ результатов моделирования. Проведенные вычислительные эксперименты подтвердили корректность и эффективность предложенной многокритериальной модели принятия решений в предсказательном методе распределения ресурсов. Результаты сценариев демонстрируют последовательную эволюцию поведения алгоритмов при усложнении условий функционирования системы.

Сценарий 1 (идеальные условия) показал, что при равномерной производительности узлов и стабильной статистике все три метода (Д, П, М) обеспечивают одинаковые результаты. Это указывает на корректность реализации алгоритмов и отсутствие искажений при базовых условиях. Парето-множество в этом случае включало все узлы, что соответствует теоретическому ожиданию – ни один из узлов не имеет преимущества при одинаковых параметрах. Таким образом, данный сценарий подтвердил адекватность исходных предпосылок и корректность механизма нормализации критериев.

Сценарий 2 (неоднородные узлы и разнотипные задачи) выявил различия между динамическим и предсказательными методами. В условиях гетерогенной производительности узлов динамический метод показал наихудшие результаты по среднему времени выполнения задач. Это объясняется тем, что он ориентируется только на моментную загрузку и не использует историческую информацию. Оба предсказательных метода (П и М) обеспечили более

сбалансированные показатели времени выполнения и загрузки узлов. Их преимущество обусловлено использованием прогнозных данных, позволяющих учитывать фактическую производительность каждого узла. При этом различия между П и М были минимальны, так как в данном сценарии достоверность прогноза оставалась высокой и влияние дополнительного критерия было незначительным.

Сценарий 3 (нестабильные условия) стал ключевым для оценки преимуществ многокритериального подхода. Введение статистических выбросов и ухудшение качества прогнозов показали, что метод (П) теряет эффективность – наблюдается рост суммарного времени выполнения. Метод (М), наоборот, демонстрирует адаптивность за счет учета достоверности прогноза: задачи перераспределяются в пользу узлов с более надежной статистикой, что снижает риск перегрузки и повышает точность прогноза. При этом наблюдается небольшое снижение равномерности загрузки, что является допустимой «платой» за повышение точности и предсказуемости выполнения задач.

Таким образом, сравнительный анализ показал, что:

1. В простых условиях все методы эквивалентны по эффективности.
2. В условиях гетерогенности преимущество имеют предсказательные методы, использующие накопленную статистику.
3. При наличии нестабильных данных только многокритериальный подход сохраняет сбалансированные показатели производительности.

Количественные результаты, представленные на графиках, подтверждают, что метод (М) обеспечивает снижение среднего времени выполнения задач на 8-10% и улучшение равномерности распределения нагрузки на 60-70% по сравнению с динамическим методом при ухудшении достоверности статистики. Это свидетельствует о практической значимости введения критерия достоверности прогноза и подтверждает его роль в обеспечении устойчивого функционирования вычислительной системы при изменяющихся условиях.

Итогом моделирования является доказательство того, что интеграция многокритериального выбора в предсказательный метод распределения ресурсов позволяет повысить качество решений и адаптивность планировщика без существенного ухудшения других показателей (равномерности нагрузки на узлы и вычислительной сложности алгоритма).

9. Заключение. В работе предложен усовершенствованный предсказательный метод распределения вычислительных ресурсов, основанный на многокритериальной модели принятия решений, который учитывает не только прогнозируемое время выполнения и загрузку узла, но и достоверность прогноза, отражающую соответствие прогнозируемых и фактических данных.

1. Построена многокритериальная модель принятия решений, определены показатели эффективности работы вычислительной системы: время обработки задач и равномерность распределения нагрузки.
2. Заданы критерии принятия решений, которые предложено расширить за счет нового, характеризующего достоверность прогнозирования.
3. Предложен подход к решению задачи многокритериальной оптимизации: *1 этап* – Парето-оптимизации; *2 этап* – выбор единственного решения на основе метода агрегирования (линейной свертки).
4. Разработан подробный пошаговый алгоритм многокритериального принятия решений, который является основным этапом предсказательного метода распределения ресурсов в вычислительных системах.
5. Проведены экспериментальные исследования показателей эффективности рассматриваемых методов распределения ресурсов. Результаты моделирования подтвердили, что при усложнении условий функционирования системы предсказательный метод с многокритериальным принятием решений обеспечивает более эффективное распределение задач, позволяя компенсировать влияние выбросов в статистике. Введение критерия достоверности позволяет улучшить выбранный показатель эффективности. Созданный инструментарий моделирования дает возможность гибкой настройки параметров для определения области целесообразного применения предложенного метода.

Предложенный предсказательный метод распределения ресурсов в вычислительных системах на основе многокритериальной модели принятия решений может быть использован как основа для создания интеллектуальных систем управления вычислительными кластерами, способных адаптироваться к изменяющимся условиям в реальном времени.

Дальнейшие исследования предполагается проводить для определения области целесообразного применения предложенного метода распределения. В частности, планируется исследовать

показатели эффективности метода в зависимости от процентного соотношения типов задач, характера поступления задач, вариативности и выбросов статистики и т.д., а также ограничения по его использованию.

Литература

1. Hussain H., Malik S.-U.-R., Hameed A., Khan S.-U., Bickler G., Min-Allah N., Qureshi M., et al. A survey on resource allocation in high performance distributed computing systems // Parallel Computing. 2013. vol. 39. no. 11. pp. 709–736. DOI: 10.1016/j.parco.2013.09.009.
2. Coleman J., Bhaskar K. PISA: An Adversarial Approach to Comparing Task Graph Scheduling Algorithms // IEEE International Parallel and Distributed Processing Symposium. 2025. pp. 54–66. DOI: 10.1109/IPDPS64566.2025.00014.
3. Topcuoglu H., Hariri S., Wu M-Y. Performance-effective and low-complexity task scheduling for heterogeneous computing // IEEE Transactions on Parallel and Distributed Systems. 2002. vol. 13. no. 3. pp. 260–274. DOI: 10.1109/71.993206.
4. Shim H. A demand-centered scheduling framework for shared supercomputing resources: modeling, metrics, and case insights // Scientific Reports. 2025. vol. 15. DOI: 10.1038/s41598-025-02353-9.
5. Свиридкин Д.О., Скороходов В.А. О задаче размещения потребителей в сетях с распределением потока. I. NP-полнота // Известия высших учебных заведений. Северо-Кавказский регион. Естественные науки. 2017. № 3-1(195-1). С. 36–41. DOI: 10.23683/0321-3005-2017-3-1-36-41.
6. Moly M.-I., Hossain Md.-A., Lecturer S., Roy O. Load Balancing Approach and Algorithm in Cloud Computing Environment // American Journal of Engineering Research (AJER). 2019. vol. 8. no. 04. pp. 99–105.
7. Клименко А.Б. Математическая модель и эвристические методы организации распределенных вычислений в системах интернета вещей // Компьютерные исследования и моделирование. 2025. Т. 17. № 5. С. 851–870. DOI: 10.20537/2076-7633-2025-17-5-851-870.
8. Wijaya C., Wiryasaputra R., Huang C.-Y., Tanato J., Yang C.-T. Load Balancing Algorithm in a Software-Defined Network Environment with Round Robin and Least Connections. Smart Grid and Internet of Things // 7th EAI International Conference. SGIoT. 2023. vol. 557. pp. 148–157. DOI: 10.1007/978-3-031-55976-1_15.
9. Albalawi N.-S. Dynamic scheduling strategies for cloud-based load balancing in parallel and distributed systems // J Cloud Comp. 2025. vol. 14. DOI: 10.1186/s13677-025-00757-6.
10. Sana M.-U., Li. Z. Efficiency aware scheduling techniques in cloud computing: a descriptive literature review // PeerJ Computer Science. 2021. vol. 7. DOI: 10.7717/peerj-cs.509.
11. Alfahid A., Lhioui C., Asklany S., et al. Peer-driven task scheduling and resource allocation for enhanced performance in industrial IoT systems // Scientific Reports. 2025. vol. 15. DOI: 10.1038/s41598-025-98910-3.
12. Li Y., Zhang Z., Ding Z. Optimization scheduling of microgrid cluster based on improved moth-flame algorithm // Energy Inform. 2024. vol. 7. DOI: 10.1186/s42162-024-00418-z.
13. Грибова В.В., Шалфеева Е.А., Филаретов В.Ф., Зуев А.В., Юхимец Д.А. Метод интеллектуального планирования миссий автономных подводных аппаратов // Информатика и автоматизация. 2025. Т. 24. № 5. С. 1257–1283. DOI: 10.15622/ia.24.5.1.

14. Писковский В.О., Лычева Е.О., Могиленец В.М. Прогнозирование временных характеристик прикладных сетевых сервисов // Вестник Московского университета. Серия № 15. Вычислительная математика и кибернетика. 2025. № 3. С. 62–73. DOI: 10.55959/MSU/0137–0782–15–2025–49–3–62–73.
15. Ламановский М.Н., Лавров Д.Н. Балансировка нагрузки облачных вычислений // Математические структуры и моделирование. 2024. № 2(70). С. 87–99. DOI: 10.24147/2222-8772.2024.2.87-99.
16. Клименко А.Б. Ресурсно-ориентированная технология организации информационного процесса распределения вычислительных ресурсов при интеграции концепций Интернета вещей и краевых вычислений // Моделирование, оптимизация и информационные технологии. 2025. № 13(3). URL: https://moitvivt.ru/ru/journal/pdf?id=2038. DOI: 10.26102/2310-6018/2025.50.3.045.
17. Sanjalawe Y., Al-Emari S., Fraihat S., et al. AI-driven job scheduling in cloud computing: a comprehensive review // Artificial Intelligence Review. 2025. vol. 58. DOI: 10.1007/s10462-025-11208-8.
18. Klimenko A., Elmekeev M. Case study of Lamarckian and Baldwin Evolution Principles Application to the Computations Planning in Resource-Constrained Ad-Hoc Networks // 18th International Conference on Management of Large-Scale System Development (MLSD). 2025. pp. 1–6. DOI: 10.1109/MLSD65526.2025.11220680.
19. Трушкин Е.С., Фрейман В.И. Предсказательный метод распределения ресурсов в вычислительных системах // Информационные технологии и вычислительные системы. 2026. № 1. С. 133–144. DOI: 10.14357/20718632260112.
20. Кулешов С.В., Зайцева А.А., Шальнев И.О. Подход к реализации распределенной системы виртуальных машин для самоорганизующихся сетей // Информационно-управляющие системы. 2019. № 5. С. 30–37. DOI: 10.31799/1684-8853- 2019-5-30-37.
21. Дубинин Р.С., Темников Д.О. Методы балансировки нагрузки между микросервисами в облачной среде // Международный научно-исследовательский журнал. 2025. № 8(158). DOI: 10.60797/IRJ.2025.158.27.
22. Клейман Л.А. Методика принятия решений в задаче диагностики элементов информационно-управляющих систем // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2021. № 38. С. 90–109. DOI: 10.15593/2224-9397/2021.2.05.
23. Barrera-Garcia J., Cisternas-Caneo F., et al. Feature Selection Problem and Metaheuristics: A Systematic Literature Review about Its Formulation, Evaluation and Applications // MDPI Biomimetics. 2024. vol. 9. no. 1. DOI: 10.3390/biomimetics9010009.
24. Idrissi A., Alaoui A. A Multi-Criteria Decision Method in the DBSCAN Algorithm for Better Clustering // International Journal of Advanced Computer Science and Applications (IJACSA). 2016. vol. 7 no. 2. pp. 377–384. DOI: 10.14569/IJACSA.2016.070252.
25. Helfrich S., Herzel A., Ruzika S., et al. Using scalarizations for the approximation of multiobjective optimization problems: towards a general theory // Math Meth Oper Res. 2023. vol. 100. pp. 27–63. DOI: 10.1007/s00186-023-00823-2.
26. Dehnad P., Bidgoli A.-A., Rahnamayan S. Beyond the Pareto Front: Utilizing the Entire Population for Decision-Making in Evolutionary Machine Learning // Mathematics. 2025. vol. 13. no. 16. DOI: 10.3390/math13162579.
27. Колесников В.Л., Бракович А.И., Жук Я.А. Решение многокритериальных задач, оптимальных по Парето // Труды БГТУ. Физико-математические науки

и информатика. 2014. № 6. URL: https://cyberleninka.ru/article/n/reshenie-mnogokriterialnyh-zadach-optimalnyh-po-pareto (дата обращения: 29.10.2025).
28. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задач: 2-е изд., испр. и доп. М.: ФИЗМАТЛИТ. 2007. 256 с.
29. Menzies T., Saint-Hilary G., Mozgunov P. A comparison of various aggregation functions in multi-criteria decision analysis for drug benefit–risk assessment // Statistical Methods in Medical Research. 2022. vol. 31. no. 5. pp. 899–916. DOI: 10.1177/09622802211072512.
30. Yang F., Li X., Liu Q., Li X., Li Z. Learning-Based Hierarchical Decision-Making Framework for Automatic Driving in Incompletely Connected Traffic Scenarios // MDPI Sensors. 2024. vol. 24. no. 8. DOI: 10.3390/s24082592.
31. Chen S., Liu J., et al. A linguistic multi-criteria decision making approach based on logical reasoning // Information Sciences. 2014. vol. 258. pp. 266–276. DOI: 10.1016/j.ins.2013.08.040.
32. Hullermeier E., Słowinski R. Preference learning and multiple criteria decision aiding: differences, commonalities, and synergies – part II // 4OR. 2024. vol. 22. pp. 313–349. DOI: 10.1007/s10288-023-00561-5.
33. Трушкин Е.С., Гаврилов А.В., Фрейман В.И. Математическое и имитационное моделирование для оценки производительности коммуникационных устройств вычислительных, информационно-управляющих и телекоммуникационных систем // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2025. № 53. С. 129–156. DOI: 10.15593/2224-9397/2025.1.07.

Трушкин Егор Сергеевич — аспирант, кафедра «Автоматика и Телемеханика», Пермский национальный исследовательский политехнический университет (ПНИПУ). Область научных интересов: вычислительные системы и информационные технологии. Число научных публикаций — 6. egor.s.trushkin@gmail.com; Комсомольский проспект, 29, 614990, Пермь, Россия; р.т.: +7(902)797-8390.

Фрейман Владимир Исаакович — д-р техн. наук, профессор, почетный работник сферы образования Российской Федерации, профессор кафедры, кафедра «Автоматика и Телемеханика», Пермский национальный исследовательский политехнический университет (ПНИПУ). Область научных интересов: вычислительные системы, информационные технологии, техническая диагностика, проектирование, управление и мониторинг телекоммуникационных систем и сетей, помехоустойчивое кодирование, цифровая обработка сигналов, теория принятия решений. Число научных публикаций — 230. vifrejman@pstu.ru; Комсомольский проспект, 29, 614990, Пермь, Россия; р.т.: +7(342)239-1816.

DOI 10.15622/ia.25.2.10

E. TRUSHKIN, V. FREYMAN

A PREDICTIVE METHOD OF RESOURCE ALLOCATION IN COMPUTING SYSTEMS BASED ON A MULTICRITERIA DECISION-MAKING MODEL

Trushkin E., Freyman V. **A Predictive Method of Resource Allocation in Computing Systems Based on a Multicriteria Decision-Making Model.**

Abstract. Modern computing systems typically operate under heterogeneous and variable load conditions. Efficient distribution of computing resources is an important tool for ensuring high performance (e.g., productivity, reliability, stability). Therefore, developing task distribution methods that can simultaneously improve several metrics is a pressing issue. The approach proposed in this paper is an extension of the predictive resource allocation method. This is achieved through the introduction of a multi-criteria decision-making model that includes the predicted execution time, the current node load, and the reliability of the forecast, assessed by the statistics of discrepancies between actual and predicted values. The object of this study is computing systems with heterogeneous nodes processing task flows of variable complexity. The subject of this study is the models and algorithms for predictive distribution of computing resources based on multi-criteria decision making. The objective of the study is to improve the efficiency and stability of heterogeneous computing systems by using a more substantiated mechanism for node selection based on a set of criteria. A review of existing dynamic and predictive distribution methods is provided, and their advantages, disadvantages, and limitations for effective application are identified. A multi-criteria decision-making model was developed that implements the construction of a set of Pareto-optimal solutions and an arbitration procedure. Software simulations were conducted under various system operating scenarios, including conditions with reduced statistical reliability. The results of the study showed that the proposed predictive method based on multi-criteria decision-making reduces the average task execution time and improves node load uniformity compared to known approaches. These results are proposed for use in the construction of heterogeneous computing systems with adaptive resource management systems.

Keywords: computing system, resource allocation methods, Pareto optimality, multi-criteria decision making, load forecasting, statistical data.

References

1. Hussain H., Malik S.-U.-R., Hameed A., Khan S.-U., Bickler G., Min-Allah N., Qureshi M., et al. A survey on resource allocation in high performance distributed computing systems. Parallel Computing. 2013. vol. 39. no. 11. pp. 709–736. DOI: 10.1016/j.parco.2013.09.009.
2. Coleman J., Bhaskar K. PISA: An Adversarial Approach to Comparing Task Graph Scheduling Algorithms. IEEE International Parallel and Distributed Processing Symposium. 2025. pp. 54–66. DOI: 10.1109/IPDPS64566.2025.00014.
3. Topcuoglu H., Hariri S., Wu M-Y. Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Transactions on Parallel and Distributed Systems. 2002. vol. 13. no. 3. pp. 260–274. DOI: 10.1109/71.993206.
4. Shim H. A demand-centered scheduling framework for shared supercomputing resources: modeling, metrics, and case insights. Scientific Reports. 2025. vol. 15. DOI: 10.1038/s41598-025-02353-9.

5. Sviridkin D., Skorokhodov V. [On the task of placing consumers in flow-distributed networks. I. NP-completeness]. Izvestiya vysshikh uchebnykh zavedeniy. Severo-Kavkazskiy region. Estestvennye nauki – News of higher educational institutions. The North Caucasus region. Natural sciences. 2017. no. 3-1(195-1). pp. 36–41. DOI: 10.23683/0321-3005-2017-3-1-36-41. (In Russ.).
6. Moly M.-I., Hossain Md.-A., Lecturer S., Roy O. Load Balancing Approach and Algorithm in Cloud Computing Environment. American Journal of Engineering Research (AJER). 2019. vol. 8. no. 04. pp. 99–105.
7. Klimenko A. [Mathematical model and heuristic methods for organizing distributed computing in Internet of Things systems]. Kompyuternye issledovaniya i modelirovanie – Computer research and modeling. 2025. vol. 17. no. 5. pp. 851–870. DOI: 10.20537/2076-7633-2025-17-5-851-870. (In Russ.).
8. Wijaya C., Wiryasaputra R., Huang C.-Y., Tanato J., Yang C.-T. Load Balancing Algorithm in a Software-Defined Network Environment with Round Robin and Least Connections. Smart Grid and Internet of Things. 7th EAI International Conference. SGIoT. 2023. vol. 557. pp. 148–157. DOI: 10.1007/978-3-031-55976-1_15.
9. Albalawi N.-S. Dynamic scheduling strategies for cloud-based load balancing in parallel and distributed systems. J Cloud Comp. 2025. vol. 14. DOI: 10.1186/s13677-025-00757-6.
10. Sana M.-U., Li. Z. Efficiency aware scheduling techniques in cloud computing: a descriptive literature review. PeerJ Computer Science. 2021. vol. 7. DOI: 10.7717/peerj-cs.509.
11. Alfahid A., Lhioui C., Asklany S., et al. Peer-driven task scheduling and resource allocation for enhanced performance in industrial IoT systems. Scientific Reports. 2025. vol. 15. DOI: 10.1038/s41598-025-98910-3.
12. Li Y., Zhang Z., Ding Z. Optimization scheduling of microgrid cluster based on improved moth-flame algorithm. Energy Inform. 2024. vol. 7. DOI: 10.1186/s42162-024-00418-z.
13. Gribova V., Shalfeeva E., Filaretov V., Zuev A., Yukhimets D. [Intelligent mission planning method for autonomous underwater vehicles]. Informatika i avtomatizatsiya – Informatics and Automation. 2025. vol. 24. no. 5. pp. 1257–1283. DOI: 10.15622/ia.24.5.1. (In Russ.).
14. Piskovsky V., Lycheva E., Mogilenets V. [Forecasting the time characteristics of applied network services]. Vestnik Moskovskogo universiteta. Seriya no. 15. Vychislitel'naya matematika i kibernetika – Bulletin of the Moscow University. Series no. 15. Computational mathematics and cybernetics. 2025. no. 3. pp. 62–73. DOI: 10.55959/MSU/0137–0782–15–2025–49–3–62–73. (In Russ.).
15. Lamanovsky M., Lavrov D. [Load balancing of cloud computing]. Matematicheskie struktury i modelirovanie – Mathematical structures and modeling. 2024. no. 2(70). pp. 87–99. DOI: 10.24147/2222-8772.2024.2.87-99. (In Russ.).
16. Klimenko A. [Resource-oriented technology for organizing the information process of computing resource allocation when integrating the concepts of the Internet of Things and edge computing]. Modelirovanie, optimizatsiya i informatsionnye tekhnologii – Modeling, optimization, and information technology. 2025. no. 13(3). DOI: 10.26102/2310-6018/2025.50.3.045. (In Russ.).
17. Sanjalawe Y., Al-Emari S., Fraihat S., et al. AI-driven job scheduling in cloud computing: a comprehensive review. Artificial Intelligence Review. 2025. vol. 58. DOI: 10.1007/s10462-025-11208-8.
18. Klimenko A., Elmekeev M. Case study of Lamarckian and Baldwin Evolution Principles Application to the Computations Planning in Resource-Constrained Ad-Hoc Networks. 18th International Conference on Management of Large-Scale System Development (MLSD). 2025. pp. 1–6. DOI: 10.1109/MLSD65526.2025.11220680.

19. Trushkin E., Freyman V. [Predictive method of resource allocation in computing systems]. Informacionnye tehnologii i vychislitel'nye sistemy – Information technologies and computing systems. 2026. no 1. pp. 133–144. DOI: 10.14357/20718632260112. (In Russ.).
20. Kuleshov S., Zaitseva A., Shalnev I. [An approach to the implementation of a distributed virtual machine system for self-organizing networks]. Informatsionno-upravlyayushchie sistemy – Information management systems. 2019. no. 5. pp. 30–37. DOI: 10.31799/1684-8853- 2019-5-30-37. (In Russ.).
21. Dubinin R., Temnikov D. [Load balancing methods between microservices in a cloud environment]. Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal – International Scientific Research Journal. 2025. no. 8(158). DOI: 10.60797/IRJ.2025.158.27. (In Russ.).
22. Kleiman L. [Decision-making methodology in the task of diagnosing elements of information management systems]. Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotekhnika, informatsionnye tekhnologii, sistemy upravleniya – Bulletin of the Perm National Research Polytechnic University. Electrical engineering, information technology, control systems. 2021. no. 38. pp. 90–109. DOI: 10.15593/2224-9397/2021.2.05. (In Russ.).
23. Barrera-Garcia J., Cisternas-Caneo F., et al. Feature Selection Problem and Metaheuristics: A Systematic Literature Review about Its Formulation, Evaluation and Applications. MDPI Biomimetics. 2024. vol. 9. no. 1. DOI: 10.3390/biomimetics9010009.
24. Idrissi A., Alaoui A. A Multi-Criteria Decision Method in the DBSCAN Algorithm for Better Clustering. International Journal of Advanced Computer Science and Applications (IJACSA). 2016. vol. 7 no. 2. pp. 377–384. DOI: 10.14569/IJACSA.2016.070252.
25. Helfrich S., Herzel A., Ruzika S., et al. Using scalarizations for the approximation of multiobjective optimization problems: towards a general theory. Math Meth Oper Res. 2023. vol. 100. pp. 27–63. DOI: 10.1007/s00186-023-00823-2.
26. Dehnad P., Bidgoli A.-A., Rahnamayan S. Beyond the Pareto Front: Utilizing the Entire Population for Decision-Making in Evolutionary Machine Learning. Mathematics. 2025. vol. 13. no. 16. DOI: 10.3390/math13162579.
27. Kolesnikov V., Brakovich A., Zhuk Ya. [Solution of multi-criteria Pareto optimal problems]. Trudy BGTU. Fiziko-matematicheskie nauki i informatika – Proceedings of BSTU. Physical and mathematical sciences and computer science. 2014. no. 6. (In Russ.).
28. Podinovskiy V., Nogin V. Pareto-optimarnye resheniya mnogokriterial'nykh zadach: 2-e izd., ispr. i dop [Pareto-optimal solutions of multi-criteria problems: 2nd ed., ispr. and add.]. Moscow: FIZMALIT, 2007. 256 p. (In Russ.).
29. Menzies T., Saint-Hilary G., Mozgunov P. A comparison of various aggregation functions in multi-criteria decision analysis for drug benefit–risk assessment. Statistical Methods in Medical Research. 2022. vol. 31. no. 5. pp. 899–916. DOI: 10.1177/09622802211072512.
30. Yang F., Li X., Liu Q., Li X., Li Z. Learning-Based Hierarchical Decision-Making Framework for Automatic Driving in Incompletely Connected Traffic Scenarios. MDPI Sensors. 2024. vol. 24. no. 8. DOI: 10.3390/s24082592.
31. Chen S., Liu J., et al. A linguistic multi-criteria decision making approach based on logical reasoning. Information Sciences. 2014. vol. 258. pp. 266–276. DOI: 10.1016/j.ins.2013.08.040.
32. Hullermeier E., Słowinski R. Preference learning and multiple criteria decision aiding: differences, commonalities, and synergies – part II. 4OR. 2024. vol. 22. pp. 313–349. DOI: 10.1007/s10288-023-00561-5.

33. Trushkin E., Gavrilov A., Freiman V. [Mathematical and simulation modeling for evaluating the performance of communication devices of computing, information management and telecommunication systems]. Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotekhnika, informatsionnye tekhnologii, sistemy upravleniya – Bulletin of the Perm National Research Polytechnic University. Electrical engineering, information technology, control systems. 2025. no. 53. pp. 129–156. DOI: 10.15593/2224-9397/2025.1.07. (In Russ.).

Trushkin Egor — Graduate student, Department of Automation and Telemechanics, Perm National Research Polytechnic University (PNRPU). Research interests: computing systems and information technology. The number of publications — 6. egor.s.trushkin@gmail.com; 29, Komsomolsky Ave., 614990, Perm, Russia; office phone: +7(902)797-8390.

Freyman Vladimir — Ph.D., Dr.Sci., Professor, Honored worker of education of the Russian Federation, Professor of the Department, Department of Automation and Telemechanics, Perm National Research Polytechnic University (PNRPU). Research interests: computing systems; information technologies; technical diagnostics; design, control and monitoring of networks; error-correcting coding; digital signal processing; decision-making. The number of publications — 230. vifrejman@pstu.ru; 29, Komsomolsky Ave., 614990, Perm, Russia; office phone: +7(342)239-1816.

Руководство для авторов

Взаимодействие автора с редакцией осуществляется через личный кабинет на сайте журнала «Информатика и автоматизация» http://ia.spcras.ru/. При регистрации авторам рекомендуется заполнить все предложенные поля данных. Подготовка статьи ведется с помощью текстовых редакторов MS Word 2007 и выше или LaTex. Объем основного текста (до раздела Литература) - от 20 до 30 страниц включительно. Переносы запрещены. Номера страниц не проставляются. Основная часть текста статьи разбивается на разделы, среди которых являются обязательными: введение, хотя бы один «содержательный» раздел и заключение. Допускается также мотивированное содержанием и структурой материала выделение подразделов. В основную часть допускается помещать рисунки, таблицы, листинги и формулы. Правила их оформления подробно изложены на нашем сайте в разделе «Руководство для авторов».

Author guidelines

Interaction between authors and the Editorial board is carried out via the personal account on the website of the journal "Informatics and Automation" http://ia.spcras.ru/. Upon registration, authors are requested to fill in all the provided data fields in the registration form. Manuscripts should be prepared using MS Word 2007 or later, or LaTex. The length of the main text (up to the References section) must be between 20 and 30 pages, inclusive. Hyphenation is not allowed. Page numbers should not be inserted. The main body of the manuscript should be divided into sections. The following sections are mandatory: an Introduction, at least one "substantive" section, and a Conclusion. The use of subsections is permitted if justified by the content and structure of the material. Figures, tables, listings, and formulas may be included in the main body. Detailed rules for their formatting are available on our website in the "Author Guidelines" section.

Подписано к печати 02.04.2026.
Формат 60×90 1/16. Усл. печ. л. 19,0. Заказ № 93. Тираж 300 экз.
Дата выхода в свет: 03.04.2026.

Отпечатано в Редакционно-издательском центре ГУАП.
Адрес типографии: Россия, 190000, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
Подписной индекс П5513 по каталогу «Почта России»

Цена свободная

Signed to print 02.04.2026
Format 60×90 1/16. Cond. printed sheets 19.0.
Order No. 93. Circulation 300 copies.
Passed for print 03.04.2026.

Printed in Publishing center GUAP.
Address: 67 Bolshaya Morskaya Street., St.Petersburg, 190000, Russia
Subscription Index П5513, Russian Post Catalog

Price not fixed