

ISSN 2713-3192
DOI 10.15622/ia.2025.24.5
<http://ia.spcras.ru>

ТОМ 24 № 5

ИНФОРМАТИКА
И АВТОМАТИЗАЦИЯ

INFORMATICS
AND AUTOMATION



СПб ФИЦ РАН

Санкт-Петербург
2025



INFORMATICS AND AUTOMATION

Volume 24 № 5, 2025

Scientific and educational journal primarily specialized in computer science, automation, robotics, applied mathematics, interdisciplinary research

Founded in 2002

Founder and Publisher

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS)

Editor-in-Chief

A. L. Ronzhin, Prof., Dr. Sci., St. Petersburg, Russia

Editorial Council

A. A. Ashimov	Prof., Dr. Sci., Academician of the National Academy of Sciences of the Republic of Kazakhstan, Almaty, Kazakhstan
I. A. Kalyaev	Prof., Dr. Sci., Academician of RAS, Taganrog, Russia
Yu. A. Merkuryev	Prof., Dr. Sci., Academician of the Latvian Academy of Sciences, Riga, Latvia
A. I. Rudskoi	Prof., Dr. Sci., Academician of RAS, St. Petersburg, Russia
V. Sgurev	Prof., Dr. Sci., Academician of the Bulgarian Academy of Sciences, Sofia, Bulgaria
B. Ya. Sovetov	Prof., Dr. Sci., Academician of RAE, St. Petersburg, Russia
V. A. Soyfer	Prof., Dr. Sci., Academician of RAS, Samara, Russia

Editorial Board

E. Azarov	Prof., Dr. Sci., Minsk, Belarus
O. Yu. Gusikhin	Ph. D., Dearborn, USA
V. Delic	Prof., Dr. Sci., Novi Sad, Serbia
A. Dolgui	Prof., Dr. Sci., St. Etienne, France
M. N. Favorskaya	Prof., Dr. Sci., Krasnoyarsk, Russia
M. Zelezny	Assoc. Prof., Ph.D., Plzen, Czech Republic
H. Kaya	Assoc. Prof., Ph.D., Utrecht, Netherlands
A. A. Karpov	Assoc. Prof., Dr. Sci., St. Petersburg, Russia
S. V. Kuleshov	Dr. Sci., St. Petersburg, Russia
A. D. Khomonenko	Prof., Dr. Sci., St. Petersburg, Russia
D. A. Ivanov	Prof., Dr. Habil., Berlin, Germany
K. P. Markov	Assoc. Prof., Ph.D., Aizu, Japan
R. V. Meshcheryakov	Prof., Dr. Sci., Moscow, Russia
N. A. Moldovian	Prof., Dr. Sci., St. Petersburg, Russia
V. V. Nikulin	Prof., Ph.D., New York, United States
V. Yu. Osipov	Prof., Dr. Sci., Deputy Editor-in-Chief, St. Petersburg, Russia
V. K. Pshikhopov	Prof., Dr. Sci., Taganrog, Russia
H. Samani	Assoc. Prof., Ph.D., Plymouth, UK
J. Savage	Assoc. Prof., Ph.D., Mexico City, Mexico
M. Secujski	Assoc. Prof., Ph.D., Novi Sad, Serbia
A. V. Smirnov	Prof., Dr. Sci., St. Petersburg, Russia
B. V. Sokolov	Prof., Dr. Sci., St. Petersburg, Russia
L. V. Utkin	Prof., Dr. Sci., St. Petersburg, Russia
L. B. Sheremetov	Assoc. Prof., Dr. Sci., Mexico, Mexico

Editor: A. S. Viktorova

Interpreter: Ya. N. Berezina

Art editor: N. A. Dormidontova

Editorial office address

SPC RAS, 39 litera A , 14-th line V.O., St. Petersburg, 199178, Russia

e-mail: ia@spcras.ru, web: <http://ia.spcras.ru>

The journal is indexed in Scopus

The journal is published under the scientific-methodological supervision of Department for Nanotechnologies and Information Technologies of the Russian Academy of Sciences

© St. Petersburg Federal Research Center of the Russian Academy of Sciences, 2025

ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ

Том 24 № 5, 2025

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

Журнал основан в 2002 году

Учредитель и издатель

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

Главный редактор

А. Л. Ронжин, д-р техн. наук, проф., Санкт-Петербург, РФ

Редакционный совет

А. А. Ашимов	академик Национальной академии наук Республики Казахстан, д-р техн. наук, проф., Алматы, Казахстан
И. А. Каляев	академик РАН, д-р техн. наук, проф., Таганрог, РФ
Ю. А. Меркурьев	академик Латвийской академии наук, д-р, проф., Рига, Латвия
А. И. Рудской	академик РАН, д-р техн. наук, проф., Санкт-Петербург, РФ
В. Сгурев	академик Болгарской академии наук, д-р техн. наук, проф., София, Болгария
Б. Я. Советов	академик РАО, д-р техн. наук, проф., Санкт-Петербург, РФ
В. А. Соифер	академик РАН, д-р техн. наук, проф., Самара, РФ

Редакционная коллегия

И. С. Азаров	д-р техн. наук, проф., Минск, Беларусь
О. Ю. Гусихин	д-р наук, Диаборн, США
В. Делич	д-р техн. наук, проф., Нови-Сад, Сербия
А. Б. Долгий	д-р наук, проф. Сент-Этьен, Франция
М. Железны	д-р наук, доцент, Пльзень, Чешская республика
Д. А. Иванов	д-р экон. наук, проф., Берлин, Германия
Х. Кайя	д-р наук, доцент, Утрехт, Нидерланды
А. А. Карпов	д-р техн. наук, доцент, Санкт-Петербург, РФ
С. В. Кулешов	д-р техн. наук, Санкт-Петербург, РФ
К. П. Марков	д-р наук, доцент, Аизу, Япония
Р. В. Мещеряков	д-р техн. наук, проф., Москва, РФ
Н. А. Молдовян	д-р техн. наук, проф., Санкт-Петербург, РФ
В. В. Никулин	д-р наук, проф., Нью-Йорк, США
В. Ю. Осипов	д-р техн. наук, проф., зам. главного редактора, Санкт-Петербург, РФ
В. Х. Пшихопов	д-р техн. наук, проф., Таганрог, РФ
Х. К. Саваж	д-р техн. наук, доцент, Мехико, Мексика
Х. Самани	д-р наук, доцент, Плимут, Соединенное Королевство
М. Сечуиски	д-р техн. наук, доцент, Нови-Сад, Сербия
А. В. Смирнов	д-р техн. наук, проф., Санкт-Петербург, РФ
Б. В. Соколов	д-р техн. наук, проф., Санкт-Петербург, РФ
Л. В. Уткин	д-р техн. наук, проф., Санкт-Петербург, РФ
М. Н. Фаворская	д-р техн. наук, проф., Красноярск, РФ
А. Д. Хомоненко	д-р техн. наук, проф., Санкт-Петербург, РФ
Л. Б. Шереметов	д-р техн. наук, Мехико, Мексика

Выпускающий редактор: А. С. Викторова

Переводчик: Я. Н. Березина

Художественный редактор: Н. А. Дормидонтова

Адрес редакции

14-я линия В.О., д. 39, лит. А, г. Санкт-Петербург, 199178, Россия

e-mail: ia@spcras.ru, сайт: <http://ia.spcras.ru>

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

© Федеральное государственное бюджетное учреждение науки

«Санкт-Петербургский Федеральный исследовательский центр Российской академии наук», 2025
Разрешается воспроизведение в прессе, а также сообщение в эфир или по кабелю опубликованных в составе печатного периодического издания - журнала «ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ» статей по текущим экономическим, политическим, социальным и религиозным вопросам с обязательным указанием имени автора статьи и печатного периодического издания журнала «ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ»

CONTENTS

Robotics, Automation and Control Systems

V. Gribova, E. Shalfeeva, V. Fialretov, A. Zuev, D. Yukhimets
THE METHOD OF INTELLIGENT MISSION PLANNING FOR AUTONOMOUS
UNDERWATER VEHICLES 1257

V. Kureichik, E. Gerasimenko
INTEGRATED ALGORITHM FOR MULTI-CRITERIA GROUP DECISION-MAKING
IN EVACUATION TASKS 1284

E. Larkin, A. Bogomolov, A. Privalov
NEURAL NETWORK TUNING OF THE GENETIC ALGORITHM FOR CONTROLLING
A VECTOR HIERARCHICAL SYSTEM 1333

D. Krasnov, M. Volynsky, A. Gusev
STATE ESTIMATION OF AN AGENT WITH STOCHASTIC DYNAMICS USING
RECURRENT FILTERS 1355

N. Viet Hung, T. Thu Huong, N. Tan, T. Cong Doan, N. Nam-Hoang
MACHINE LEARNING APPLICATIONS FOR DELIVERY TIME PREDICTION AND
FREIGHT PLANNING 1379

Artificial Intelligence, Knowledge and Data Engineering

A.A. Kokate, T.R. Jadhav
A NOVEL APPROACH TO EEG ARTIFACT REMOVAL USING ADASYN
AND OPTIMIZED HIERARCHICAL 1D CNN 1408

A. Fedotova, A. Romanov
TECHNIQUE FOR IDENTIFYING TEXTS GENERATED BY LARGE LANGUAGE
MODELS 1444

Information Security

A. Balyabin, S. Petrenko
MODEL OF A CRITICAL INFORMATION INFRASTRUCTURE CLOUD PLATFORM
WITH CYBER IMMUNITY 1471

P. Borisov, D. Varlamov, Yu. Kosolapov
A GRAPH SIMILARITY CALCULATION ALGORITHM AND ITS APPLICATION
FOR COMPARING BINARY EXECUTABLE FILES 1506

СОДЕРЖАНИЕ

Робототехника, автоматизация и системы управления

В.В. Грибова, Е.А. Шалфеева, В.Ф. Филаретов, А.В. Зув, Д.А. Юхимец
МЕТОД ИНТЕЛЛЕКТУАЛЬНОГО ПЛАНИРОВАНИЯ МИССИЙ АВТОНОМНЫХ ПОДВОДНЫХ АППАРАТОВ 1257

В.В. Курейчик, Е.М. Герасименко
ИНТЕГРИРОВАННЫЙ АЛГОРИТМ МНОГОКРИТЕРИАЛЬНОГО ГРУППОВОГО ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ ЗАДАЧ ЭВАКУАЦИИ 1284

Е.В. Ларкин, А.В. Богомолов, А.Н. Привалов
НЕЙРОСЕТЕВАЯ НАСТРОЙКА ГЕНЕТИЧЕСКОГО АЛГОРИТМА УПРАВЛЕНИЯ ВЕКТОРНОЙ ИЕРАРХИЧЕСКОЙ СИСТЕМОЙ 1333

Д.И. Краснов, М.А. Волынский, А.А. Гусев
ОЦЕНКА СОСТОЯНИЯ АГЕНТА С ДИНАМИКОЙ СТОХАСТИЧЕСКОГО ХАРАКТЕРА С ПОМОЩЬЮ РЕКУРРЕНТНЫХ ФИЛЬТРОВ 1355

Н.В. Хунг, Т.Т. Хуонг, Н. Тан, Т.К. Доан, Н. Нам-Хоанг
ПРИЛОЖЕНИЯ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ СРОКОВ ДОСТАВКИ И ПЛАНИРОВАНИЯ ПЕРЕВОЗОК 1379

Искусственный интеллект, инженерия данных и знаний

А.А. Кокате, Т.Р. Джадхав
НОВЫЙ ПОДХОД К УДАЛЕНИЮ АРТЕФАКТОВ ЭЭГ С ИСПОЛЬЗОВАНИЕМ ADASYN И ОПТИМИЗИРОВАННОЙ ИЕРАРХИЧЕСКОЙ ОДНОМЕРНОЙ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ (1D CNN) 1408

А.М. Федотова, А.С. Романов
МЕТОДИКА ИДЕНТИФИКАЦИИ ТЕКСТОВ, СГЕНЕРИРОВАННЫХ БОЛЬШИМИ ЯЗЫКОВЫМИ МОДЕЛЯМИ 1444

Информационная безопасность

А.А. Балябин, С.А. Петренко
МОДЕЛЬ ОБЛАЧНОЙ ПЛАТФОРМЫ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С КИБЕРИММУНИТЕТОМ 1471

П.Д. Борисов, Д.В. Варламов, Ю.В. Косолапов
АЛГОРИТМ ВЫЧИСЛЕНИЯ ПОХОЖЕСТИ ГРАФОВ И ЕГО ПРИМЕНЕНИЕ ДЛЯ СРАВНЕНИЯ БИНАРНЫХ ИСПОЛНЯЕМЫХ ФАЙЛОВ 1506

В.В. ГРИБОВА, Е.А. ШАЛФЕЕВА, В.Ф. ФИЛАРЕТОВ, А.В. ЗУЕВ,
Д.А. ЮХИМЕЦ
**МЕТОД ИНТЕЛЛЕКТУАЛЬНОГО ПЛАНИРОВАНИЯ МИССИЙ
АВТОНОМНЫХ ПОДВОДНЫХ АППАРАТОВ**

Грибова В.В., Шалфеева Е.А., Филаретов В.Ф., Зувев А.В., Юхимец Д.А. Метод интеллектуального планирования миссий автономных подводных аппаратов.

Аннотация. Создание полностью автономных необитаемых подводных аппаратов и комплексов, способных выполнять различные исследовательские и технологические операции в условиях неопределенности, является актуальной задачей. Ключевой проблемой является автоматическая коррекция миссий в реальном времени на основе данных от бортовых систем. Целью данной работы является разработка метода интеллектуального планирования миссий на стратегическом уровне управления автономными подводными робототехническими комплексами, обеспечивающего автоматическое формирование адаптивных планов и их преобразование в исполнительные команды тактического уровня для работы в изменяющихся условиях среды. В статье авторами определены принципы разработки интеллектуального планировщика миссий автономных подводных робототехнических комплексов (АПРК) на стратегическом уровне и менеджера миссий для управления миссией на тактическом уровне с формированием конкретных заданий исполнителям. Разработана формальная модель планирования миссий через множество линейных участков с предусловиями и постусловиями. Ключевым аспектом предложенного решения является использование онтологического подхода для стандартизации описания миссий и обеспечения их программной интерпретации. Создана специализированная среда разработки миссий на облачной платформе IASaaS, позволяющая экспертам формировать и адаптировать планы миссий без углубления в технические детали. Разработан комплекс инструментальных средств с модульной архитектурой, обеспечивающий масштабируемость и адаптацию решения для различных классов АПРК и типов миссий. Результаты апробации показали, что предложенное решение позволяет формировать гибкие планы, учитывающие разнообразие ситуаций, и автоматически выбирать последовательности команд в зависимости от поступающих данных. Полученные результаты открывают новые возможности для создания полностью автономных подводных комплексов, способных выполнять сложные исследовательские и технологические операции без постоянного контроля оператора. Дальнейшие исследования направлены на совершенствование алгоритмов менеджера миссий, а также на интеграцию планировщика с другими компонентами бортового обеспечения.

Ключевые слова: автономный подводный аппарат, робототехнический комплекс, интеллектуальное планирование, миссия, неопределённость.

1. Введение. Важным направлением развития в области применения автономных необитаемых подводных аппаратов является их оснащение новым оборудованием и усложнение выполняемых ими задач [1 – 3].

Одной из ключевых проблем в создании полностью автономных подводных робототехнических комплексов (АПРК) является необходимость автоматического формирования и коррекции

в реальном времени действий роботов на основе поставленной общей цели [4 – 7] и информации от бортовых систем об изменяющихся условиях. Для решения этой проблемы при работе в сложных и неопределенных условиях рабочей среды необходима разработка информационно-управляющих систем (ИУС) нового поколения – с элементами искусственного интеллекта.

ИУС современных роботов имеют многоуровневую структуру [8]: поступающая в нее информация относится к разным уровням «иерархии». Предполагается, что если ИУС имеет *«стратегический»* (высший) *уровень* управления автономным аппаратом или роботом, то он должен отвечать за модельные представление робота об окружающей рабочей среде и за принятие решений об общем плане выполнения полученных заданий с учетом априорных и накопленных знаний об этой среде, конфигурации и состоянии бортового оборудования. При этом общий план действий на этом уровне за счет использования интеллектуального планировщика миссий (ИПМ) должен расчленяться на конкретные элементарные операции, с выбором их последовательности и параметров выполнения. Следующий – *тактический»* *уровень* ИУС должен преобразовывать выбранные элементарные операции в желаемые позиции аппарата или, например, его манипулятора, в скорости или усилиях, которые они должны развивать. Это позволяет сформировать требуемые законы согласованного изменения движения АПРК. Для отработки этих программных законов *исполнительный»* *уровень* управления вырабатывает конкретные воздействия на соответствующие движители и приводы манипулятора с учетом обратных связей, подающих соответствующие сигналы от датчиков внутренней информации о состоянии исполнительной системы АПРК.

В работе [9] рассмотрены вопросы создания архитектуры интеллектуальной многоагентной системы планирования и формирования заданий для автономных необитаемых подводных аппаратов, в основу которой положен алгоритм планирования Graphplan [10]. Особенностью созданной системы является использование базы знаний поведения автономных аппаратов при решении типовых задач в сочетании с сетевой организацией взаимодействия интеллектуальных агентов планирования. При этом авторами разработан механизм опосредованной централизованной коммуникации агентов, реализуемый на принципах «доски объявлений». Полученный механизм позволяет разделить основной процесс поиска на непосредственно процессы решений и управления,

взаимодействия между которыми совершаются посредством механизма взаимодействия агентов.

В качестве серьезного недостатка описанной интеллектуальной системы планирования и формирования заданий для подводных аппаратов необходимо отметить отсутствие эффективного механизма формирования базы знаний экспертами предметной области, а также сложность автономной реализации решателя интеллектуальной системы на борту. Кроме того, предложенный метод не учитывает возможность установки сложного дополнительного оборудования, в частности, многозвенных манипуляторов.

Разработан метод планирования миссии для группы подводных аппаратов [11], который предполагает оптимальное распределение задач между ними в соответствии с их характеристиками (бортовым оборудованием и автономностью). Указанный метод основан на модификации алгоритма RRT (Rapidly-exploring Random Tree Star), который позволяет спланировать маршрут каждого робота в зависимости от заданных показателей эффективности и особенностей окружающей среды (препятствия, течения) так, чтобы гарантировать выполнение всех целей, поставленных в миссии. Однако в качестве целей миссии может быть сформировано только достижение заданных целевых маршрутных точек, что не позволяет использовать описанный метод для планирования миссий с более сложными целями, которые предполагают выполнение технологических операций, поиск и детальное обследование заранее неизвестных объектов интереса.

В работе [12] предлагается метод построения планировщика миссий для группы беспилотных летательных аппаратов. Указанный планировщик состоит из двух основных компонентов: генератор миссий, который формирует возможные варианты действий для каждого аппарата, и модуль помощника, который оценивает сгенерированные планы действий с точки зрения достижимости целей, эффективности и затрат ресурсов. Однако генерация новых планов действий и выбор плана на запуск обеспечивается непосредственно оператором, на которого возлагаются интеллектуальные функции планировщика. При этом отсутствие высокоскоростных каналов связи не позволяет эффективно включить оператора в процесс перепланирования миссии, так как он не получит необходимого объема данных для принятия решений.

Методы автоматического планирования миссий в настоящее время наиболее актуальны в аэрокосмической области при создании автономных космических аппаратов и роверов [13], предназначенных для исследования поверхности других планет. При их создании

основным требованием является возможность перепланирования миссии в зависимости от изменения текущей обстановки без вмешательства оператора. При этом само планирование в большинстве случаев сводится только к планированию маршрутов робота на основе статистического анализа возможных ситуаций и выбора оптимального варианта. Кроме того, запуск новой миссии на исполнение часто требует прямого подтверждения оператора.

В работе [14] описана архитектура информационно-управляющей системы необитаемого аппарата с интеллектуальным алгоритмом принятия решений, обеспечивающая автоматическое выполнение исследовательских миссий в подводной среде с неизвестными препятствиями. Предложенный алгоритм принятия решений учитывает информацию об окружающей среде, параметрах заданной миссии и составе бортового оборудования. При этом архитектура менеджера, управляющего миссией, реализуется в реальном времени с использованием сетей Петри.

Анализ представленной работы показал, что предложенный алгоритм принятия решений не предполагает автоматическое выполнение комплексных миссий для поиска заданных объектов работы и выполнения различных манипуляционных операций.

Проблемы реализации «тактического» и «исполнительного» уровней ИУС АПРК в большинстве уже практически решены [3, 4, 15, 16]. При этом исследования по созданию ИУС с элементами искусственного интеллекта для различных роботов активно развиваются и к настоящему времени уже создано множество отдельных довольно эффективных подходов: для разработки алгоритмов автономной навигации, для решения SLAM задач (одновременная локализация и картографирование – Simultaneous Localization and Mapping) [17 – 19], классификации объектов и оптимизации планирования маршрутов в сложной среде [20 – 22], алгоритмы распределения задач между группами роботов [23, 24], анализа информации, непрерывно получаемой от датчиков в реальном времени [25, 26], для диагностики блоков и устройств [27] и др.

Созданные решения уже позволяют выполнять довольно сложные задачи в подводной среде, однако полная автономность и интеллектуальность АПРК остается еще не достигнутой из-за большой сложности и неопределенности подводной среды, меняющихся условий функционирования этих АПРК, большого разнообразия решаемых ими задач, необходимости принятия решений в условиях неполных данных или неожиданных событий. Отдельно стоит отметить технические особенности, связанные с ограниченными

вычислительными ресурсами и отсутствием большого количества данных для обучения в отличие от других типов роботизированных устройств.

Указанные факторы требуют создания новых подходов, направленных на развитие интеллектуальных средств управления и ИУС АПРК на стратегическом уровне.

Целью данной работы является разработка метода интеллектуального планирования миссий на стратегическом уровне управления автономными подводными робототехническими комплексами, обеспечивающего автоматическое формирование адаптивных планов и их преобразование в исполнительные команды тактического уровня для работы в изменяющихся условиях среды.

2. Теоретические аспекты планирования миссии в условиях неопределенности. Решение сложных интеллектуальных задач обычно предполагает последовательное выполнение ряда этапов. Для задачи планирования этапами являются:

- формулировка четко определенной цели;
- разработка плана миссии (пошаговой стратегии достижения цели);
- реализация/выполнение действий, непосредственно направленных на ее достижение.

В контексте АПРК цели определяются спецификой их применения, включая научные исследования, военные операции, поисково-спасательные миссии, экологический мониторинг и технологическое использование. При этом возможности такого применения вытекают из (или ограничиваются) множеством известных команд АПРК.

При разработке планов миссий и определении действий АПРК, направленных на достижение целей, необходимо учитывать множество факторов, связанных с возможными сценариями развития событий в процессе выполнения задачи, характеристики операционной зоны, а также технические параметры самого АПРК.

Особое внимание следует уделить последовательной передаче команд через управляющую систему АПРК, что является критически важной особенностью реализации операций. Поэтому при разработке плана миссии требуется определить *потенциальное множество линейных планов действий*, выбор из которого будет происходить, исходя из текущей обстановки.

Обозначим четко определенную цель (желаемое, достигаемое целевое состояние) как s_{goal} , множество известных *действий-команд* – как $\{a_j\}$, *предусловие* j -го действия (условие на состояние, в котором

его можно применять) и его *постусловие* (условие на состояние, возникающее после выполнения действия) – как $\langle \varphi_{pred}^j, \varphi_{post}^j \rangle$ соответственно. Тогда s_{goal} – это одно из состояний процесса-сценария, другое очевидное состояние – начальное, s_0 . При этом каждое состояние (начальное, целевое или состояние, возникающее после выполнения одного из действий) может определяться набором аспектов $\{asp^i\}$: местоположение относительно других сущностей, собственная конфигурация АПРК, свойства его «частей», время, полученные сведения и др. В некоторых простых сценариях или заданиях все состояния могут быть перечислены ($s_0, s_1, s_2, \dots, s_M, s_{goal} \in S$) и именованы.

Необходимо учитывать и такую особенность реализации/выполнения *действий-команд*, как их передача на АПРК (через управляющую систему АПРК) строго последовательно. При этом выполняться они могут параллельно, т.к. исполнители – части единого робототехнического комплекса.

Введем термин *линейный участок* (последовательная группа действий) – это *действия-команды*, выстроенные друг за другом $\langle a_1, a_2, \dots, a_T \rangle$, которые передаются строго по очереди. Тогда помимо предусловий и постусловий $\langle \varphi_{pred}^j, \varphi_{post}^j \rangle$ действий a_j следует обращаться и к предусловиям/постусловиям *групп действий* $d_{jk}: \langle \varphi_{pred}^{jk}, \varphi_{post}^{jk} \rangle$.

При разработке плана типовой миссии требуется определить потенциальное множество таких *групп действий-команд* – линейных участков. На этапе исполнения выбор *групп действий-команд* из этого множества должен производиться, исходя из некоторых условий, обстановки (текущего состояния).

Принимая для простоты вариант, когда все состояния могут быть перечислены, постановку *задачи планирования* абстрактной (типовой) миссии можно выразить так.

Дано:

$$s_0, s_{goal} \{a_j = \langle \varphi_{pred}^j, \varphi_{post}^j \rangle\}.$$

Найти:

потенциальное множество *линейных участков* планов (*групп действий-команд*) $\{\langle d_1, \dots, d_n \rangle\}$, $d_k = \langle a_{ik}, \dots, a_{mk} \rangle$ с определенными $\langle \varphi_{pred}^k, \varphi_{post}^k \rangle$ для каждого $d_k \in D$ и условий их применения – правил $\{s_{jk} \rightarrow d_{jk}\}$, обеспечивающих пошаговую, поэтапную стратегию достижения цели. Т.е. обеспечивается существование не менее одной цепочки от s_0 к s_{goal} : $\langle s_0 = \varphi_{pred}^1, s_1 = \varphi_{post}^1 \rangle, \langle s_1 = \varphi_{pred}^2, s_2 =$

$\varphi_{post}^3\rangle, \dots, \langle s_{k-1} = \varphi_{pred}^{1k}, s_{goal} = \varphi_{post}^{k+1}\rangle$ (т.е. в такой цепочке постусловие каждой *группы* совпадает с предусловием другой *группы* в этой цепочке или с целью).

Это множество пар, сопоставляющих условию применения группы действий $\{s_{jk} \rightarrow d_{jk}\}$, является базой знаний о возможностях и вариантах реагирования АПРК на разные ситуации [9, 10] при выполнении типовой задачи.

Существование множества цепочек от s_0 к s_{goal} (в плане типовой миссии) дает шанс гарантировать выполнение всех целей, поставленных в миссии [11], если не будет препятствий (конкретных и общего характера), предусмотренных планом.

Отметим, что в типовом плане могут быть заложены варианты оптимизации работ или маршрута с учетом многообразия аспектов изменяющейся обстановки, особенно при рассмотрении спектра возможных результатов выполнения (постусловий) отдельных действий в составе группы. По результатам команд могут вырабатываться сообщения о событиях $\{ev_i\}$.

В общем случае *действие-команда* a_k имеет параметры выполнения $p_{k1}, \dots, p_{kn}$. Некоторым параметрам могут быть назначены значения на уровне типового задания (например, минимальная скорость), некоторые могут быть наделены значениями при конкретизации места работы или исполняющих устройств (например, высота над поверхностью при поиске гидролокатором), некоторые – смогут получить значения в реальной обстановке в зависимости от достигнутых успехов или проблем (например, оценка оставшегося заряда). С некоторыми командами могут быть связаны сообщения о событиях с параметрами $ev_i(p_1, \dots, p_{kn})$, например, сообщение о найденном объекте и его координатах.

Правило $\{\{s_k \rightarrow d_k\}\}$ должно рассматриваться детально. Левая часть правила описывает условия по разным аспектам состояния процесса, что может быть выражено через события (оповещения или сигналы), измерения или вычисления. Состояние в текущий момент t есть набор свершившихся событий или значений их параметров и текущие значения свойств, атрибутов, конфигураций $s_k = \{ev_i(t, p_{1l}, \dots, p_{im}) | asp_j(t)\}^i$. Правая часть правила может требовать применения вычислительных функций для всех или некоторых параметров команд из d_k : $a_{k1}(p_{1l}, \dots, p_{1n}), \dots, a_{kl}(p_{1l}, \dots, p_{1m})$. Группа команд есть последовательность команд с возможностью их итеративного выполнения.

При этом $asp_j(t)$ имеет свою (j -ую) область возможных значений, например, диапазон возможной скорости движения, а любой параметр p_j может $\in \{asp^{i,t}\}$, например, скорость, с которой надо приблизиться к объекту для захвата, а для получения значения параметра команды может потребоваться применение функции или операции из $\{f_1, f_2, \dots, f_Q\}$, например, для расчета расстояния до указанной точки пространства. Для нетривиальных вычислений и для удобства связывания параметров команд с актуальными значениями могут применяться переменные $V = \{v_1, v_2, \dots, v_P\}, f_q: V_n \rightarrow V$. Поэтому функции или операции для вычисления актуальных величин являются важной частью адаптируемых планов.

Планирование абстрактной миссии как набора правил для множества линейных участков представляет собой этап, выполняемый человеком-оператором и обладающий наибольшей интеллектуальной сложностью. Метод данного планирования заключается в декомпозиции сценария достижения цели на необходимый и достаточный набор правил выбора действий (подзадач), зависящих от развития ситуации в реальных условиях. Информационное обеспечение процесса планирования ограничивается потоком событий, включающим: сообщения о возникновении проблемных ситуаций; данные от систем технического зрения и сенсоров; сигналы успешного выполнения командных последовательностей. Указанные классы событий формируют предусловия для активации соответствующих правил выбора действий, обеспечивая тем самым адаптивность плана к изменяющимся условиям операционной среды.

При этом у эксперта предметной области есть возможность предусмотреть типичные особенности окружающей среды (препятствия, течения) и описать типичные реакции на них [11]. В качестве целей миссии может быть сформировано не только достижение заданных целевых маршрутных точек, а более сложные цели, включая выполнение технологических операций, поиск и детальное обследование объектов интереса, в том числе заранее неизвестных, выполнение исследовательских миссий с неизвестными препятствиями, выполнение комплексных миссий, предполагающих поиск заданных объектов и выполнение различных манипуляционных операций [14].

На современном этапе *планирование абстрактной миссии* может быть поддержано подсказками и контролем полноты всех возможных состояний и непротиворечивости правил.

Общий, типовой план миссии адаптируется к реальному выполнению задачи (в смысле физического использования аппарата

для очередного достижения типовой цели) через задание специфических характеристик АПРК, особенностей территорий, на которых они выполняются, прочих объектов и субъектов миссии (которые содержатся в s_0 и s_{goal}).

Этап конкретизации (адаптирования s_0 и s_{goal} к s'_0 и s'_{goal}) позволяет сформировать вариант плана типовой задачи с учетом заданных показателей эффективности [11].

Конкретизация может быть поддержана оценением плана действий с точки зрения достижимости целей, эффективности и затрат ресурсов [12].

Задачу реализации (т.е. выбора действий для выполнения) можно выразить так:

Дано:

момент t ,

$\{ev_i(t | p_{1l}, \dots, p_{im})\}$,

$\{asp_j(t)\}$;

база знаний поведения АПРК $\{s_k \rightarrow d_k\}$ (с предвычисленными значениями тех параметров, которые зависели от s'_0 и s'_{goal}).

Найти:

очередной участок плана $d_k = \langle a_{ik}, \dots, a_{mk} \rangle$, верный относительно базы знаний (в предположении, что в базе для каждой комбинации $\{s_{ik}\}$, $s_{ik} = ev_i | asp_{ik}$ есть одно правило $s_j \rightarrow d_j$).

Этот этап может быть поручен программному исполнителю, когда уже построена пошаговая стратегия достижения цели. Программа-супервизор будет использовать базу знаний при принятии решения о выборе очередного участка линейного плана в конкретной ситуации.

3. Идея разделения планирования типового (шаблонного) и уникального сценария миссии. Планы миссий обладают свойством повторного использования. С одной стороны, для разных существующих (и определяемых) целей очевидны однотипные действия и их последовательности/совокупности.

С другой стороны, для одной цели могут сложиться разные условия выполнения, более того, могут формулироваться заведомо универсальные цели для их отработки в разных местах и временах. Это связано с идеей адаптировать разработанный план миссии к выполнению различных задач (конкретным робототехническим комплексом для очередного достижения типовой цели). Каждая такая конкретная миссия уточняет общий план миссии через задание специфических характеристик АПРК, особенностей территорий, на которых они выполняются, прочих объектов и субъектов миссии.

Метод декомпозиции сценария достижения цели на набор правил выбора действий предполагает наличие формализованного языка и средств для структурированного описания полного стратегического плана выполнения миссии. Данный подход учитывает ограниченность информации о состоянии процесса, предоставляет механизмы для вычисления динамически изменяемых параметров и обеспечивает точное преобразование в команды исполнительного уровня с параметрами.

В соответствии с вышеприведенными постановками задачи важны:

- язык и средство универсального планирования миссии экспертами предметной области, которые бы обеспечивали и облегчали создание и редактирование планов миссий с учетом специфики возникающих задач и условий их выполнения,
- метод и средство конкретизации универсального плана к специфике объектов и субъектов миссии и условий выполнения,
- метод и средство доведения до исполнителя (или алгоритма) локального выбора команд и других действий конкретного плана, а также их конкретизации путем назначения текущих значений параметрам.

Высокоуровневое планирование миссии в соответствии с описанием задания с учетом ресурсов АПРК, работоспособности его компонентов и внешних условий принято относить к стратегическому уровню (информационно-управляющей системы). Локальное планирование, которое адаптирует глобальные задачи к конкретным условиям работы отдельного устройства, принято относить к тактическому уровню; это планирование действий для текущей ситуации в рамках типовой запланированной миссии.

На основании изложенного предлагается общий подход к разработке системы планирования миссий, который отделяет работу тактического уровня от стратегического так, чтобы каждый вид планирования выполнялся в своих условиях, но с требуемым соответствием друг другу. Решено структурировать и стандартизировать описания миссий обоих уровней, создав для этого единую онтологию планирования миссий АПРК.

Использование онтологического подхода означает:

- 1) предоставление человеку семантического шаблона для формирования плана стратегического уровня, использование справочников и каталогов, систематизирующих свойства и номенклатуру сущностей этой предметной области;

2) обеспечение максимального соответствия при переводе конкретизируемого плана в семантический шаблон тактического уровня;

3) программную интерпретируемость плана тактического уровня (универсальным обработчиком миссий на борту).

Для описания всех составных частей планов предложены семантические шаблоны, которые в свою очередь связываются друг с другом отношениями вложенности, чередования и причинности.

Для исходного состояния s_0 – это узел «Исходные данные задания», содержащий набор узлов «параметр»;

для ожидаемых состояний s^{jk} и s_{goal} – узлы «конфигурация оборудования» и «Условия», внутри которого узлы «событие», «исключение» и узел «Значения переменных и состояние» с соответствующими внутренностями для ограничения диапазонов значений или их вычислений через функции или операции. Для действий $d_k = \langle a_{ik}, \dots, a_{mk} \rangle$, приводящих к смене состояний, – узлы «Присваивание значения», «Команда» (с «параметрами») и конструктор итераций (повторений), связывающий параметр некоторой «Команды» внутри с элементами из «конечное множество для параметра цикла».

Семантические шаблоны стратегического уровня в настоящее время являются основой создания специализированной среды разработки на онтологических платформах.

В рамках идеи разделения средств универсального планирования миссии от средств локального планирования действий в текущей ситуации предоставляется развитый инструментарий редактирования планов. Инструменты создания и редактирования планов миссий позволяют формировать и уточнять их с учетом специфики задач и условий их выполнения.

Такой подход позволяет структурировать и стандартизировать описание планов миссий в соответствии с общими принципами. Возможность создания универсального менеджера миссий на борту АПРК (без необходимости перепрограммирования для каждой новой миссии) обещает повысить эффективность планирования и выполнения миссий, снизить временные и ресурсные затраты.

Формирование типового плана миссии осуществляется экспертами предметной области в специализированной среде разработки миссий. На этом (стратегическом) уровне среда планирования будет включать и обеспечение полноты и корректности данных и параметров для требуемых в миссии команд, и контроль потенциальных результатов исполнения команд. На тактическом

уровне решаются подзадачи обхода препятствий, учета динамических изменений окружающей среды и др.

4. Архитектура комплексной системы планирования миссий с элементами искусственного интеллекта. Для описанного трехшагового планирования миссий с адаптивностью работы и автономностью подводного робототехнического комплекса предлагается интегрированное решение, объединяющее стратегическое, тактическое и исполнительное управление с использованием элементов искусственного интеллекта.

Для обеспечения отмеченного ряда этапов деятельности планирования миссии (таких как *формулировка абстрактной цели и разработка плана пошагового достижения цели* в условиях неопределенности, *конкретизация цели и реализация действий, направленных на ее достижение*) предлагается информационно-управляющая система, включающая рабочие места эксперта и инженера с интеллектуальной поддержкой, и программный супервизор с генератором групп конкретных команд (участков плана), имитирующий умные решения на основе способности интерпретировать декларативные знания (рисунок 1).



Рис. 1. Архитектура системы планирования миссий АПРК

Основой ИУС являются база знаний о возможностях и вариантах реагирования АПРК на разные ситуации при выполнении типовой задачи и программный менеджер миссий, который умеет читать текущую ситуацию и руководствоваться этой базой для выбора команд на исполнение и перевода их на требуемый исполнителю язык.

Набор правил является результатом передачи «стратегического» плана, адаптированного к текущей исходной ситуации, на тактический уровень, а менеджер миссий передает результат ее интерпретации на исполнительный уровень (рисунок 2).

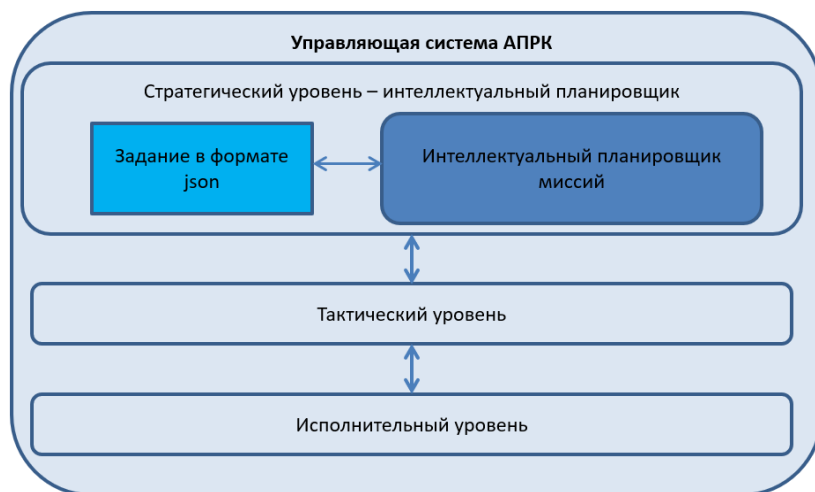


Рис. 2. Схема управляющей системы АПРК

Планы действий размещаются в трех информационных компонентах: в библиотеке миссий (где хранятся типовые планы), конкретный набор правил, и текущая плановая цепочка. Для взаимной преемственности (и повторной используемости) создана онтология предметной области – структура (язык) описания миссии (листинг 1) с соглашениями по ее применению.

Онтология определяет структуру записи знаний и другой информации (как семантический шаблон), правила их формирования знаний и интерпретации. Семантический шаблон – необходимый элемент декларативного представления данных и знаний. В основу декларативного представления положено двухуровневое представление информации, а для построения самой онтологии

используется язык описания метаинформации [28, 29]. Реализация данного решения выполнена на платформе IACPaas (<https://iacpaas.dvo.ru>) [28].

```

План задания АПРК {СПИСОК}
  Исходные данные задания {СПИСОК}
    параметр задания
    конфигурация оборудования
  Подмиссия (Правило) {СПИСОК}
    пояснение (Строковое)
  Условия {СПИСОК} ('сору')
    Сообщения о событиях*{СПИСОК}
      Сообщение о событии (Строковое)
      исключение(Строковое)
      нет сигналов(Строковое)
    Значения переменных и состояние процесса
      состояние процесса
    Требуемые вычисления {СПИСОК}
      вычислительная функция {СПИСОК}
        входной параметр
        ожидаемый результат
      код выполнения
    Присваивание значения{СПИСОК}
  Блок с действиями{СПИСОК}
    Команда{СПИСОК} ('list')
      исполнитель(Строковое)
    входные параметры команды ('set')
    Результаты{СПИСОК}
      ожидаемый ответ
      возможное исключение
    повторение {СПИСОК}
      конечное множество для цикла
      Команда
    Ожидаемое изменение состояния процесса
  
```

Листинг 1. Модель онтологии описания и хранения миссии

Информационный компонент системы, содержащий знание о вариантах реагирования конкретного АПРК на разные ситуации при выполнении запланированной задачи, обеспечивает процесс принятия решений в реальном времени. Этот компонент создается в соответствии с онтологией. Семантические шаблоны записи типовых и конкретизированных планов миссий идентичны. Такая архитектура позволяет классифицировать соответствующий программный интерпретатор (менеджер миссий) как онтолого-ориентированный. Учитывая, что данный модуль осуществляет прием входных данных от

менеджера команд и генерирует упорядоченные командные последовательности для исполнительных устройств, то должен быть реализован для развертывания непосредственно на борту автономного комплекса.

Система планирования включает *среду разработки миссий, библиотеку миссий, среду описания задания, среду создания системных библиотек*. Ключевые компоненты взаимодействуют через четко определенные интерфейсы. Использование JSON в качестве формата передачи данных обеспечивает универсальность, читаемость и легкую интеграцию с различными подсистемами (рисунок 2).

Среда разработки миссий включает специализированный редактор описания планов, функционирующий на основе единой онтологии, что обеспечивает стандартизацию и совместимость создаваемых планов. Важной особенностью редактора является возможность абстрагирования от технических деталей, позволяющая экспертам сосредоточиться на высокоуровневом описании задач и их логической последовательности. Эта среда предоставляет экспертам предметной области интуитивно понятный инструментарий для формирования и редактирования планов миссий (рисунок 3). Созданные планы миссий сохраняются в *библиотеке миссий*, которая служит централизованным репозиторием для хранения и повторного использования ранее разработанных общих сценариев. Среда обеспечивает эффективный механизм поиска и выбора подходящих планов, а для адаптации выбранного под конкретные условия выполнения миссии имеется *среда описания задания*.

Среда создания системных библиотек поддерживает формирование набора вспомогательных информационных ресурсов:

- оборудование и характеристики АПРК, включая динамические и технические параметры,
- стандартизированные наборы команд и событий, согласованные с разработчиками программного обеспечения тактического и стратегического уровней,
- вычислительные операции и алгоритмы обработки данных, используемые в ходе выполнения миссий.

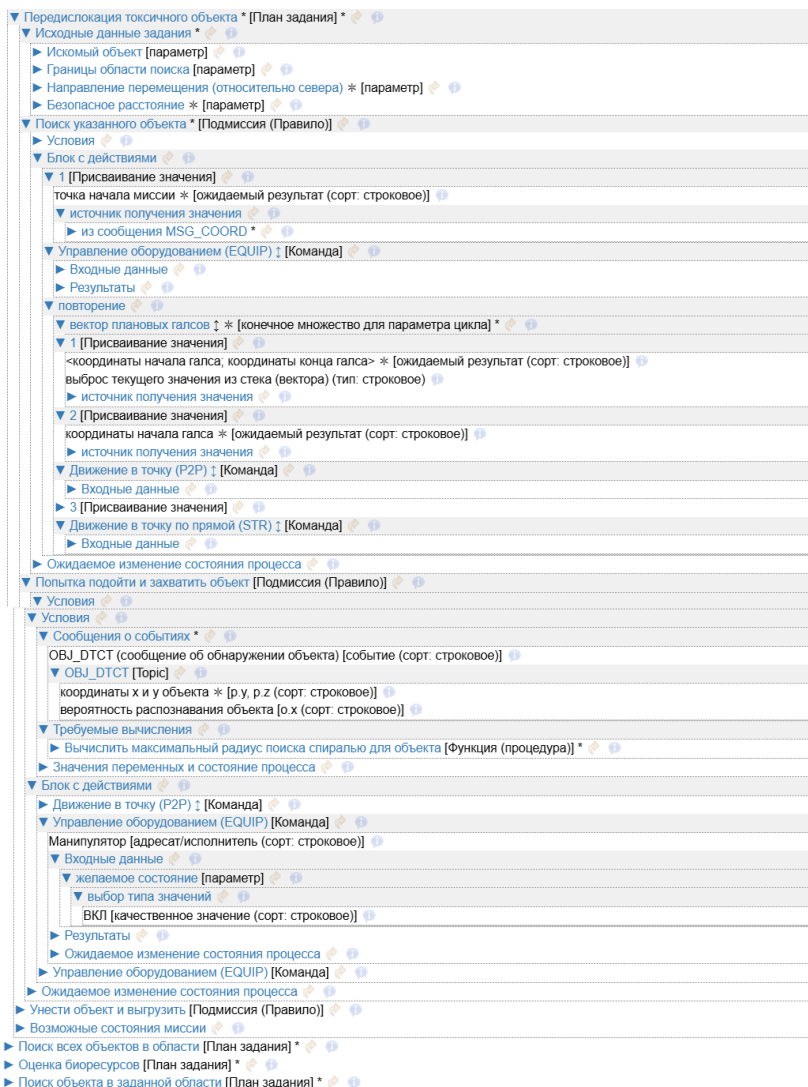


Рис. 3. Скриншот библиотеки описания типовых миссий на платформе IACPaaS

Среда описания задания предназначена для детализации параметров выполнения конкретного задания. В этой среде оператор задает:

- точные технические характеристики задействованных модулей АПРК (двигателей, систем технического зрения, манипуляторов и др.);
- географические координаты и параметры рабочей зоны;
- дополнительные условия выполнения миссии.

Результатом работы в этой среде является задание (экспортируемое в формат JSON), содержащее:

- выбранный план миссии из библиотеки,
- уточненные технические параметры АПРК,
- детальные характеристики рабочей среды.

Система планирования миссий использует облачную платформу IASPaas [29], которая обеспечивает возможность формирования знаний в структурированном и интерпретируемом виде и поддерживает коллективную разработку.

ИУС реализует *трехуровневую модель управления*, обеспечивающую гибкость и адаптивность выполнения миссий. На *стратегическом уровне* производится интеллектуальное планирование типовых миссий с контролем полноты и выполнимости; проводится адаптация глобальных задач к локальным условиям выполнения с анализом полученного задания и выполнимости при исходном состоянии ресурсов АПРК.

На *тактическом уровне* менеджер миссий анализирует полученное конкретное задание, выбирает плановые действия, применимые при текущем состоянии ресурсов АПРК и известных внешних условиях. Он оценивает достижимость подцелей, принимает решения о корректировке маршрута или миссии или о ее досрочном завершении при возникновении критических ситуаций, отвечает за координацию работы подсистем для выполнения текущих задач. Тем самым менеджер миссий связывает стратегический уровень с тактическим уровнем.

Исполнительный уровень, представленный менеджером команд, обеспечивает непосредственное выполнение сформированных команд (движение, манипуляции, сбор данных), мониторинг состояния систем АПРК, обратную связь с вышестоящими уровнями управления.

Онтологический подход важен для стандартизации форматов данных, возможности повторного использования планов миссий и их легкой адаптации к параметрам выполнения задач.

Предусмотренный механизм поиска и выбора планов в библиотеке типовых планов (которые затем адаптируются под конкретные условия выполнения миссии) значительно сокращает время подготовки новых заданий и повышает согласованность различных миссий.

5. Метод реализации среды планирования миссии и среды симуляции выполнения плана в условиях возникающих ситуаций. Менеджер миссий имеет модули для приема и обработки сообщений и для выполнения инструкций.

Модуль приема и обработки сообщений (рисунок 4):

- обеспечивает интеграцию с ROS (Robot Operating System),
- реализует механизмы подписки на топики и фильтрации сообщений,
- обеспечивает корректную интерпретацию поступающих данных в контексте выполняемой миссии.

Модуль выполнения инструкций:

- проводит проверку условий перехода между этапами миссии,
- выполняет необходимые вычисления для уточнения параметров выполнения,
- формирует команды для тактического уровня на основе текущего состояния системы,
- обеспечивает механизмы обратной связи и корректировки плана выполнения.

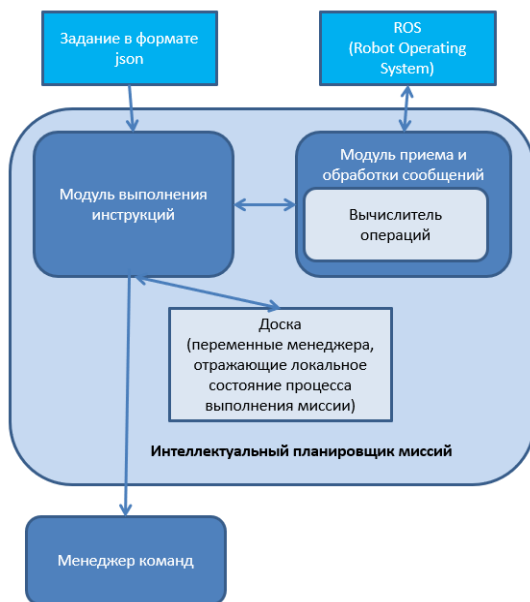


Рис. 4. Модель ИПМ на тактическом уровне

Пример работы Модуля приема и обработки сообщений в среде симуляции после получения конкретных значений s_0 : по выбранной начальной подмиссии вычисляется вектор координат-точек (с учетом ширины гидролокатора бокового обзора (и размера его перекрытия на встречных курсах) и выдается команда перехода в начальную точку. Симулируется обход галсами по заданным координатам. Проверяется, что общая длина галсов не больше максимальной дальности аппарата (рисунок 5). Приходит сообщение от гидролокатора бокового обзора об обнаружении объекта, его координатах и проценте достоверности. Проверяется, что процент достоверности больше заданного (80%), и запускается подмиссия, в которой при прибытии в точку обнаружения включается камера и запускается поиск по спирали. Приходит сообщение об обнаружении объекта камерой.

```

ub24-04@ub24-04-VMware-Virtual-Platform:~/anpa_msg$ ros2 run node_m PsSub
Миссия - Галс
Миссия - Идти в точку
Миссия - Подъем объекта
Вектор события - [['Галс', ['Начальная'], '', '']]
Вектор события - [['Идти в точку', ['Галс'], 'Pose', 'obj_dtct_gbo']]
Вектор события - [['Подъем объекта', ['Обход найденных точек'], 'Pose', 'obj_dtct_cam']]
=====> Миссия Галс
вычислить fv_gals
Результат функции [[100.0, 112.0], [200.0, 112.0], [200.0, 136.0], [100.0, 136.0], [100.0, 160.0], [200.0, 160.0], [200.0, 184.0], [100.0, 184.0]]
вычислить Path_Lenght
Результат функции 2384.0
Условие Проверка ограничения на макс путь
2384.0 < 8222.0
Результат вычисления условия: True
Подписчик и Публишер на Pose и Vector3 активированы
=====> Init
> > > > > > Pose ( 100.0 112.0 10.0 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 1.0 1.0 0.0 0.0 0.0 0.0 7.0 )
> > > > > > Pose ( 200.0 112.0 10.0 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 200.0 136.0 10.0 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 100.0 136.0 10.0 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 100.0 160.0 10.0 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 200.0 160.0 10.0 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 200.0 184.0 10.0 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 100.0 184.0 10.0 1.3 1.1 0.0 1.0 )
< < < Pose ( 126.1665 125.645652 65.1354 91.0 22.2 31.1 1.0 ) Topik obj_dtct_gbo
Обрабатываем сообщение POSE
ИД объекта 1.0 Координаты 126.17 125.65 65.14 Вероятность 91.00 %
Нашли [[126.1665, 125.645652, 65.1354, 91.0]]
Запуск миссии Идти в точку
=====> Миссия Идти в точку
Условие Порог процентов
81.0 > 80.0
Результат вычисления условия: True
> > > > > > Pose ( 126.1665 125.645652 9.99 1.3 1.1 0.0 1.0 )
> > > > > > Pose ( 2.0 1.0 0.0 0.0 0.0 0.0 7.0 )
> > > > > > Pose ( 2.88 8.22 65.1354 9.3 3.0 0.0 5.0 )
< < < Pose ( 126.5 125.2 65.14 91.0 22.2 31.1 1.0 ) Topik obj_dtct_cam
Обрабатываем сообщение POSE
ИД объекта 1.0 Координаты 126.50 125.20 65.14 Вероятность 91.00 %

```

Рис. 5. Фрагмент работы менеджера миссий

5. Заключение. В работе предложены метод, принципы создания и архитектура комплекса информационных и программных компонентов интеллектуального планировщика миссий автономных подводных робототехнических комплексов. Разработанный метод обеспечивает ряд ключевых преимуществ: достижение высокой степени стандартизации благодаря онтологическому подходу; повышение гибкости и адаптивности за счёт реализации трехуровневой системы управления; возможность полностью автономного функционирования в условиях динамически изменяющейся операционной обстановки; существенное сокращение временных затрат на подготовку новых миссий; широкую адаптируемость решения к конкретным требованиям различных типов миссий и классов АПРК.

Важнейшим элементом системы стала единая онтология планирования миссий, использующая семантические шаблоны стратегического уровня и составляющая основу специализированной среды разработки. Метод стандартизованного описания планов миссий в соответствии с данной онтологией гарантирует их совместимость, непротиворечивость и возможность многократного повторного использования в разнообразных операционных контекстах, что в совокупности способствует созданию качественно нового уровня автономности подводных робототехнических систем.

Литература

1. Aldhaheiri S., Masi G.D., Pairet E., Ardón P. Underwater Robot Manipulation: Advances, Challenges and Prospective Ventures // *Proceedings of the OCEANS 2022 – Chennai*. 2022. pp. 1–7. DOI: 10.1109/OCEANSChennai45887.2022.9775489.
2. Sahoo A., Dwivedy S.K., Robi P.S. Advancements in the field of autonomous underwater vehicle // *Ocean Engineering*. 2019. vol. 181. pp. 145–160. DOI: 10.1016/j.oceaneng.2019.04.011.
3. Antonelli G. *Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems*. Berlin, Heidelberg: Springer, 2003. 268 p.
4. Филаретов В.Ф., Зуев А.В., Тимошенко А.А. Особенности выполнения технологических операций с помощью автономных необитаемых подводных аппаратов, оснащаемых многозвенными манипуляторами // *Вестник Дальневосточного отделения Российской академии наук*. 2024. № 3(235). С. 165–177.
5. Azar A.T., Koubaa A. *Artificial Intelligence for Robotics and Autonomous Systems Applications*. Springer Cham. 2023. 486 p.
6. Kirchner F., Straube S., Kühn D., Hoyer N. *AI Technology for Underwater Robots*. Springer Cham. 2020. 193 p.
7. Машошин А.И. Технологии искусственного интеллекта в задачах управления автономным необитаемым подводным аппаратом // *Мехатроника, автоматизация, управление*. 2022. Т. 23. № 11. С. 596–606.

8. Петров А.А. Активное формирование моделей проблемной среды оцувствленными роботами. М.: Институт проблем передачи информации РАН, 1997. 229 с.
9. Раговский А.П. Интеллектуальная динамическая система формирования заданий для автономных необитаемых подводных аппаратов // Проблемы развития корабельного вооружения и судового радиоэлектронного оборудования. 2015. № 4. С. 100–108.
10. Blum A., Furst M. Fast planning through planning graph analysis // Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95). Montreal, Canada. 1995. pp. 1636–1642.
11. Liu F., Xu W., Feng Z., Yu C., Liang X., Su Q., Gao J. Task Allocation and Path Planning Method for Unmanned Underwater Vehicles // Drones. 2025. vol. 9. no. 6. DOI: 10.3390/drones9060411.
12. Maier S., Kiam J., Schulte A. Adaptive Mission Planning: Evaluation of a Hybrid Cognitive Mixed-Initiative Planning Assistant in Manned-Unmanned Teaming Operations // Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2024. pp. 3492–3499. DOI: 10.1109/SMC54092.2024.10831141.
13. Cunningham T., Spencer D. Automated Onboard Mission Planning for Robust and Flexible Rover Operations // Proceedings of the IEEE Aerospace Conference (AERO). 2022. pp. 1–19. DOI: 10.1109/AERO53065.2022.9843729.
14. Bian X., Chen T., Yan Z., Qin Z. Autonomous mission management and intelligent decision for AUV // Proceedings of the IEEE International Conference on Mechatronics and Automation. 2009. pp. 2101–2106. DOI: 10.1109/ICMA.2009.5246027.
15. Simetti E., Campos R., Vito D.D., Quintana J., Antonelli G., Garcia R., Turetta A. Sea Mining Exploration With an UVMS: Experimental Validation of the Control and Perception Framework // IEEE/ASME Transactions on Mechatronics. 2021. vol. 26. no. 3. pp. 1635–1645. DOI: 10.1109/TMECH.2020.3025973.
16. Филаретов В.Ф., Юхимец Д.А. Особенности синтеза высокоточных систем управления скоростным движением и стабилизацией подводных аппаратов в пространстве. Владивосток: Дальнаука, 2016. 400 с.
17. Chen L., Liu Y., Dong P., Liang J., Wang A. An Intelligent Navigation Control Approach for Autonomous Unmanned Vehicles via Deep Learning-Enhanced Visual SLAM Framework // IEEE Access. 2023. vol. 11. pp. 119067–119077. DOI: 10.1109/ACCESS.2023.3326754.
18. Lin J., Liang T., Ma G., Li B., Yu T. Research on SLAM Intelligent Robot Based on Visual Laser Fusion // Proceedings of the 8th International Conference on Control, Robotics and Cybernetics (CRC). 2024. pp. 53–57. DOI: 10.1109/CRC60659.2023.10488663.
19. Борейко А.А., Инзарцев А.В., Машошин А.И., Павин А.М., Пашкевич И.В. Система управления АНПА большой автономности на базе мультиагентного подхода // Подводные исследования и робототехника. 2019. № 2(28). С. 23–31.
20. Ferri G., Faggiani A., Celi F., Tesi A., Been R. Exploiting Behaviour Trees in Underwater Autonomous Robotic Networks // Proceedings of the OCEANS 2024 – Singapore. 2024. pp. 1–10. DOI: 10.1109/OCEANS51537.2024.10682233.
21. Han S., Zhao J., Li X., Yu J., Wang S., Liu Z. Online Path Planning for AUV in Dynamic Ocean Scenarios: A Lightweight Neural Dynamics Network Approach // IEEE Transactions on Intelligent Vehicles. 2024. vol. 9. no. 2. pp. 3782–3795. DOI: 10.1109/ITV.2024.3356529.
22. Chatterjee I., Zalte S. Machine Learning Applications: From Computer Vision to Robotics. Wiley-IEEE Press. 2023. 240 p.

23. Guo J., Li D., He B. Intelligent Collaborative Navigation and Control for AUV Tracking // IEEE Transactions on Industrial Informatics. 2021. vol. 17. no. 3. pp. 1732–1741. DOI: 10.1109/TII.2020.2994586.
24. Liu X.-F., Fang Y., Zhan Z.-H., Jiang Y.-L., Zhang J. A Cooperative Evolutionary Computation Algorithm for Dynamic Multiobjective Multi-AUV Path Planning // IEEE Transactions on Industrial Informatics. 2024. vol. 20. no. 1. pp. 669–680. DOI: 10.1109/TII.2023.3268760.
25. Guo Y., Fang X., Dong Z., Mi H. Research on multi-sensor information fusion and intelligent optimization algorithm and related topics of mobile robots // EURASIP J. Adv. Signal Process. 2021. vol. 2021. DOI: 10.1186/s13634-021-00817-4.
26. Li X., Xu S. Multi-Sensor Complex Network Data Fusion Under the Condition of Uncertainty of Coupling Occurrence Probability // IEEE Sensors Journal. 2021. vol. 21. no. 22. pp. 24933–24940. DOI: 10.1109/JSEN.2021.3061437.
27. Инзарцев А.В., Грибова В.В., Клещев А.С. Интеллектуальная система для формирования адекватного поведения автономного подводного робота в аварийных ситуациях // Подводные исследования и робототехника. 2015. № 2(20). С. 4–11.
28. Gribova V., Moskalenko P., Timchenko V., Shalfeyeva E. Intelligent Services Development Technology Using the IACPaaS Cloud Platform // Communications in Computer and Information Science. 2022. vol. 1625. pp. 19–38. DOI: 10.1007/978-3-031-15882-7_2.
29. Gribova VV, Moskalenko PM, Timchenko VA, Shalfeyeva EA. The IACPaaS Platform for Developing Systems Based on Ontologies: A Decade of Use. Scientific and Technical Information Processing. 2023. vol. 50(5). pp. 406–413. DOI: 10.3103/S0147688223050064.

Грибова Валерия Викторовна — д-р техн. наук, член-корреспондент РАН, зам. директора по научной работе, ИАПУ ДВО РАН. Область научных интересов: семантическое моделирование, онтологический инжиниринг, технологии создания гибридных интеллектуальных систем. Число научных публикаций — 400. gribova@iacp.dvo.ru; улица Радио, 5, 690041, Владивосток, Россия; р.т.: +7(423)231-3999.

Шалфеева Елена Арефьевна — д-р техн. наук, ведущий научный сотрудник, лаборатория интеллектуальных систем имени А.С. Клещева, ИАПУ ДВО РАН. Область научных интересов: семантическое моделирование, объяснительный искусственный интеллект, онтологический инжиниринг, технология программирования, технология создания систем с декларативными базами знаний. Число научных публикаций — 120. shalf@dvo.ru; улица Радио, 5, 690041, Владивосток, Россия; р.т.: +7(423)231-0424.

Филаретов Владимир Федорович — заведующий лабораторией, лаборатория робототехнических систем, ИАПУ ДВО РАН. Область научных интересов: робототехника, мехатроника, управление сложными динамическими объектами, адаптивное управление. Число научных публикаций — 600. filaret@iacp.dvo.ru; улица Радио, 5, 690041, Владивосток, Россия; р.т.: +7(423)231-0439.

Зуев Александр Валерьевич — д-р техн. наук, доцент, старший научный сотрудник, лаборатория робототехнических систем, ИАПУ ДВО РАН. Область научных интересов: робототехника, мехатроника, адаптивное управление, отказоустойчивое управление. Число научных публикаций — 200. zuev@iacp.dvo.ru; улица Радио, 5, 690041, Владивосток, Россия; р.т.: +7(423)231-0439.

Юхимец Дмитрий Александрович — д-р техн. наук, доцент, ведущий научный сотрудник, лаборатория робототехнических систем, ИАПУ ДВО РАН. Область научных интересов: робототехника, групповое управление, робастное управление. Число научных публикаций — 55. undim@iacp.dvo.ru; улица Радио, 5, 690041, Владивосток, Россия; р.т.: +7(423)231-0439.

Поддержка исследований. Исследование выполнено за счет гранта Российского научного фонда № 25-41-00044 (<https://rscf.ru/project/25-41-00044>).

V. GRIBOVA, E. SHALFEEVA, V. FIALRETOV, A. ZUEV, D. YUKHIMETS
**THE METHOD OF INTELLIGENT MISSION PLANNING FOR
AUTONOMOUS UNDERWATER VEHICLES**

Gribova V., Shalfeeva E., Fialretov V., Zuev A., Yukhimets D. The Method of Intelligent Mission Planning for Autonomous Underwater Vehicles.

Abstract. Creation of fully autonomous unmanned underwater vehicles and systems capable of performing various research and technological operations under uncertainty conditions is a pressing issue. The key problem is automatic mission correction in real time based on data from onboard systems. The aim of this work is to develop a method for intelligent mission planning at the strategic control level of autonomous underwater robotic systems, enabling automatic generation of adaptive plans and their transformation into tactical-level executable commands for operation in changing environmental conditions. In the article, the authors define the principles of developing an intelligent mission planner for autonomous underwater robotic systems (AURS) at the strategic level and a mission manager for managing the mission at the tactical level with the formation of specific tasks for performers. A formal model for planning missions through many linear sections with preconditions and postconditions has been developed. The key aspect of the proposed solution is the use of an ontological approach to standardize the description of missions and ensure their software interpretation. A specialized mission development environment has been created on the IACPaaS cloud platform, allowing experts to form and adapt mission plans without delving into technical details. The set of tools with a modular architecture has been developed, ensuring scalability and adaptation of the solution for various classes of submarines and types of missions. The results of testing have shown that the proposed solution allows for the formation of flexible plans that take into account the diversity of situations and automatically select command sequences depending on the incoming data. The results obtained open up new possibilities for creating fully autonomous underwater systems capable of performing complex research and technological operations without constant operator control. Further research is aimed at improving the mission manager algorithms, as well as integrating the planner with other onboard support components.

Keywords: autonomous underwater vehicle, robotic complex, intelligent planning, mission, uncertainty.

References

1. Aldhaheri S., Masi G.D., Pairet E., Ardón P. Underwater Robot Manipulation: Advances, Challenges and Prospective Ventures. Proceedings of the OCEANS 2022 – Chennai. 2022. pp. 1–7. DOI: 10.1109/OCEANSChenai45887.2022.9775489.
2. Sahoo A., Dwivedy S.K., Robi P.S. Advancements in the field of autonomous underwater vehicle. Ocean Engineering. 2019. vol. 181. pp. 145–160. DOI: 10.1016/j.oceaneng.2019.04.011.
3. Antonelli G. Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems. Berlin, Heidelberg: Springer, 2003. 268 p.
4. Filaretov V.F., Zuev A.V., Timoshenko A.A. [Features of performing technological operations using autonomous uninhabited underwater vehicles equipped with multi-link manipulators]. Vestnik Dalnevostochnogo otdeleniia Rossiiskoi akademii nauk – Bulletin of the Far Eastern Branch of the Russian Academy of Sciences. 2024. no. 3(235). pp. 165–177. (In Russ.).

5. Azar A.T., Koubaa A. Artificial Intelligence for Robotics and Autonomous Systems Applications. Springer Cham. 2023. 486 p.
6. Kirchner F., Straube S., Kühn D., Hoyer N. AI Technology for Underwater Robots. Springer Cham. 2020. 193 p.
7. Mashoshin A.I. [Artificial intelligence technologies in the control tasks of an autonomous uninhabited underwater vehicle]. Mekhatronika, avtomatizatsiia, upravlenie –Mechatronics, automation, control. 2022. vol. 23. no. 11. pp. 596–606.
8. Petrov A.A. Aktivnoe formirovanie modeli problemnoi sredy ochuvstvlennymi robotami [Active formation of problem environment models by sensitive robots]. Moscow: Institute of Information Transmission Problems of the Russian Academy of Sciences, 1997. 229 p. (In Russ.).
9. Ragovskii A.P. [An intelligent dynamic task generation system for autonomous uninhabited underwater vehicles]. Problemy razvitiia korabelnogo voozruzheniia i sudovogo radioelektronnogo oborudovaniia – Problems of development of ship armament and ship radio-electronic equipment. 2015. no. 4. pp. 100–108. (In Russ.).
10. Blum A., Furst M. Fast planning through planning graph analysis. Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95). Montreal, Canada. 1995. pp. 1636–1642.
11. Liu F., Xu W., Feng Z., Yu C., Liang X., Su Q., Gao J. Task Allocation and Path Planning Method for Unmanned Underwater Vehicles. Drones. 2025. vol. 9. no. 6. DOI: 10.3390/drones9060411.
12. Maier S., Kiam J., Schulte A. Adaptive Mission Planning: Evaluation of a Hybrid Cognitive Mixed-Initiative Planning Assistant in Manned-Unmanned Teaming Operations. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2024. pp. 3492–3499. DOI: 10.1109/SMC54092.2024.10831141.
13. Cunningham T., Spencer D. Automated Onboard Mission Planning for Robust and Flexible Rover Operations. Proceedings of the IEEE Aerospace Conference (AERO). 2022. pp. 1–19. DOI: 10.1109/AERO53065.2022.9843729.
14. Bian X., Chen T., Yan Z., Qin Z. Autonomous mission management and intelligent decision for AUV. Proceedings of the IEEE International Conference on Mechatronics and Automation. 2009. pp. 2101–2106. DOI: 10.1109/ICMA.2009.5246027.
15. Simetti E., Campos R., Vito D.D., Quintana J., Antonelli G., Garcia R., Turetta A. Sea Mining Exploration With an UVMS: Experimental Validation of the Control and Perception Framework. IEEE/ASME Transactions on Mechatronics. 2021. vol. 26. no. 3. pp. 1635–1645. DOI: 10.1109/TMECH.2020.3025973.
16. Filaretov V.F., Iukhimets D.A. Osobennosti sinteza vysokotochnykh sistem upravleniia skorostnym dvizheniem i stabilizatsiei podvodnykh apparatov v prostranstve [Features of synthesis of high-precision control systems for high-speed movement and stabilization of underwater vehicles in space]. Vladivostok: DalnaukaPubl., 2016. 400 p. (In Russ.).
17. Chen L., Liu Y., Dong P., Liang J., Wang A. An Intelligent Navigation Control Approach for Autonomous Unmanned Vehicles via Deep Learning-Enhanced Visual SLAM Framework. IEEE Access. 2023. vol. 11. pp. 119067–119077. DOI: 10.1109/ACCESS.2023.3326754.
18. Lin J., Liang T., Ma G., Li B., Yu T. Research on SLAM Intelligent Robot Based on Visual Laser Fusion. Proceedings of the 8th International Conference on Control, Robotics and Cybernetics (CRC). 2024. pp. 53–57. DOI: 10.1109/CRC60659.2023.10488663.
19. Boreiko A.A., Inzartcev A.V., Mashoshin A.I., Pavin A.M., Pashkevich I.V. [AUV control system with high autonomy based on a multi-agent approach]. Podvodnye issledovaniia i robototekhnika – Underwater research and robotics. 2019. no. 2(28). pp. 23–31.

20. Ferri G., Faggiani A., Celi F., Tesi A., Been R. Exploiting Behaviour Trees in Underwater Autonomous Robotic Networks. *Proceedings of the OCEANS 2024 – Singapore*. 2024. pp. 1–10. DOI: 10.1109/OCEANS51537.2024.10682233.
21. Han S., Zhao J., Li X., Yu J., Wang S., Liu Z. Online Path Planning for AUV in Dynamic Ocean Scenarios: A Lightweight Neural Dynamics Network Approach. *IEEE Transactions on Intelligent Vehicles*. 2024. vol. 9. no. 2. pp. 3782–3795. DOI: 10.1109/TIV.2024.3356529.
22. Chatterjee I., Zalte S. *Machine Learning Applications: From Computer Vision to Robotics*. Wiley-IEEE Press. 2023. 240 p.
23. Guo J., Li D., He B. Intelligent Collaborative Navigation and Control for AUV Tracking. *IEEE Transactions on Industrial Informatics*. 2021. vol. 17. no. 3. pp. 1732–1741. DOI: 10.1109/TII.2020.2994586.
24. Liu X.-F., Fang Y., Zhan Z.-H., Jiang Y.-L., Zhang J. A Cooperative Evolutionary Computation Algorithm for Dynamic Multiobjective Multi-AUV Path Planning. *IEEE Transactions on Industrial Informatics*. 2024. vol. 20. no. 1. pp. 669–680. DOI: 10.1109/TII.2023.3268760.
25. Guo Y., Fang X., Dong Z., Mi H. Research on multi-sensor information fusion and intelligent optimization algorithm and related topics of mobile robots. *EURASIP J. Adv. Signal Process*. 2021. vol. 2021. DOI: 10.1186/s13634-021-00817-4.
26. Li X., Xu S. Multi-Sensor Complex Network Data Fusion Under the Condition of Uncertainty of Coupling Occurrence Probability. *IEEE Sensors Journal*. 2021. vol. 21. no. 22. pp. 24933–24940. DOI: 10.1109/JSEN.2021.3061437.
27. Inzartsev A.V., Gribova V.V., Kleshchev A.S. [An intelligent system for the formation of adequate behavior of an autonomous underwater robot in emergency situations]. *Podvodnye issledovaniia i robototekhnika – Underwater research and robotics*. 2015. no. 2(20). pp. 4–11. (In Russ.).
28. Gribova V., Moskalenko P., Timchenko V., Shalfeyeva E. Intelligent Services Development Technology Using the IACPaaS Cloud Platform. *Communications in Computer and Information Science*. 2022. vol. 1625. pp. 19–38. DOI: 10.1007/978-3-031-15882-7_2.
29. Gribova VV, Moskalenko PM, Timchenko VA, Shalfeyeva EA. The IACPaaS Platform for Developing Systems Based on Ontologies: A Decade of Use. *Scientific and Technical Information Processing*. 2023. vol. 50(5). pp. 406–413. DOI: 10.3103/S0147688223050064.

Gribova Valerya — Ph.D., Dr.Sci., Corresponding member of the Russian Academy of Sciences, Deputy director for research, IACP FEB RAS. Research interests: semantic modeling, ontological engineering, technologies for creating hybrid intelligent systems. The number of publications — 400. gribova@iacp.dvo.ru; 5, Radio St., 690041, Vladivostok, Russia; office phone: +7(423)231-3999.

Shalfeyeva Elena — Ph.D., Dr.Sci., Leading researcher, A.S. Kleshchev laboratory of intelligent systems, IACP FEB RAS. Research interests: semantic modeling, explanatory artificial intelligence, ontological engineering, programming technology, technology for creating systems with declarative knowledge bases. The number of publications — 120. shalf@dvo.ru; 5, Radio St., 690041, Vladivostok, Russia; office phone: +7(423)231-0424.

Fialretov Vladimir — Head of the laboratory, Laboratory of robotic systems, IACP FEB RAS. Research interests: robotics, mechatronics, control of complex dynamic objects, adaptive control. The number of publications — 600. fialret@iacp.dvo.ru; 5, Radio St., 690041, Vladivostok, Russia; office phone: +7(423)231-0439.

Zuev Aleksandr — Ph.D., Dr.Sci., Associate Professor, Senior researcher, Laboratory of robotic systems, IACP FEB RAS. Research interests: robotics, mechatronics, adaptive control, fault-tolerant control. The number of publications — 200. zuev@iacp.dvo.ru; 5, Radio St., 690041, Vladivostok, Russia; office phone: +7(423)231-0439.

Yukhimets Dmitry — Ph.D., Dr.Sci., Associate Professor, Leading researcher, Laboratory of robotic systems, IACP FEB RAS. Research interests: robotics, group control, robust control. The number of publications — 55. undim@iacp.dvo.ru; 5, Radio St., 690041, Vladivostok, Russia; office phone: +7(423)231-0439.

Acknowledgements. The research was supported by the Russian Science Foundation, grant No. 25-41-00044 (<https://rscf.ru/project/25-41-00044>).

В.В. КУРЕЙЧИК, Е.М. ГЕРАСИМЕНКО
**ИНТЕГРИРОВАННЫЙ АЛГОРИТМ
МНОГОКРИТЕРИАЛЬНОГО ГРУППОВОГО ПРИНЯТИЯ
РЕШЕНИЙ ДЛЯ ЗАДАЧ ЭВАКУАЦИИ**

Курейчик В.В., Герасименко Е.М. Интегрированный алгоритм многокритериального группового принятия решений для задач эвакуации.

Аннотация. В данной статье представлен интегрированный алгоритм многокритериального группового принятия решений на основе интуиционистского нечеткого гибридного оператора усреднения для выбора оптимальной стратегии эвакуации. Данный алгоритм является многоуровневым, где на первом уровне осуществляется построение модели задачи в виде эвакуационной транспортной сети. На втором уровне производится ранжирование помещений для эвакуации для нахождения их оптимального порядка на основе различных критериев, таких как вместимость помещения, легкость эвакуации, уровень безопасности помещения как убежища, время организации эвакуации из этого места и удаленность от источника опасности. Ранжирование помещений для эвакуации осуществляется на основе многокритериального группового принятия решений и интуиционистского нечеткого гибридного оператора усреднения для моделирования сомнений и неуверенности экспертов для оценки критериев эвакуации, эвакуационных альтернатив и важности экспертов. Алгоритм оперирует лингвистическими оценками экспертов, позволяя на их основе вычислять веса экспертов и критериев для эффективного принятия решения. Агрегирование оценок осуществляется на основе модифицированного алгоритма, который позволяет оперировать весами критериев, представленными в виде нечетких интуиционистских значений, в отличие от традиционных четких чисел, на основе разработанных модифицированных операторов возведения нечетких чисел в нечеткую степень для учета сомнений и неуверенности группы экспертов. На третьем уровне осуществляется макрокопическая потоковая динамическая эвакуация с учетом возможности размещения пострадавших в помещениях для эвакуации, не являющихся убежищами. Достоинством предложенного алгоритма является возможность моделирования транспортировки эвакуируемых в динамических условиях из опасных зон с учетом их размещения в промежуточных пунктах для максимизации числа спасенных жизней, учитывая неопределённость окружающей среды, нечёткий характер факторов, влияющих на принятие эвакуационных решений, неуверенность и сомнения группы экспертов при оценке эвакуационных стратегий. Для подтверждения эффективности разработанного алгоритма было проведено моделирование эвакуации и создана программная среда, реализованная на языке JavaScript. Проведено сравнение разработанного алгоритма принятия решения на основе оператора IFNA, оперирующего нечеткими весами критериев, с существующими алгоритмами и сделан вывод о достоверности предлагаемого алгоритма. Проведена оценка зависимости времени работы алгоритма от размера входных, подтверждающая возможность использования предложенного алгоритма для больших зданий и транспортных сетей.

Ключевые слова: нечеткое интуиционистское принятие решений, нечеткая макрокопическая эвакуация, нечёткие операторы агрегирования.

1. Введение. С древних времен чрезвычайные ситуации (ЧС) приводили к катастрофическим последствиям и уносили сотни жизней.

Более того, ЧС могут возникать неожиданно, тем самым увеличивая потери и ущерб. Такие внезапные аварии часто связаны с отсутствием планов реагирования на ЧС. По сфере возникновения ЧС подразделяются на природные, экологические и техногенные [1] (рисунок 1).

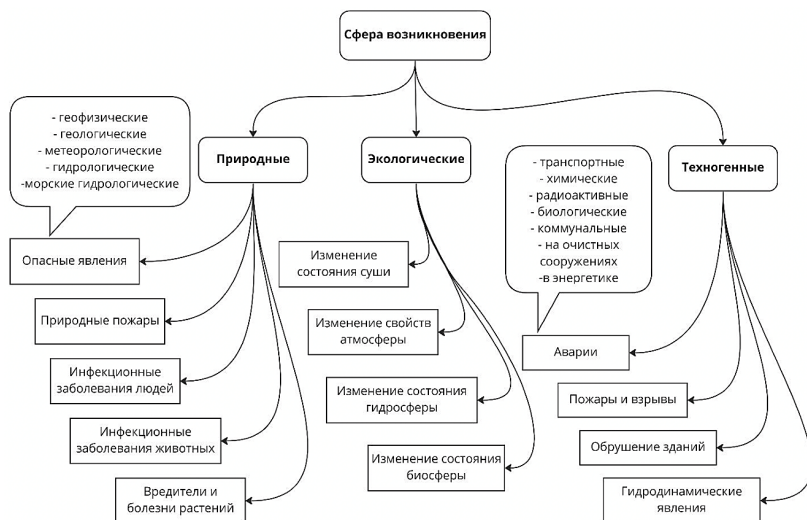


Рис. 1. Укрупненная классификация ЧС по сфере возникновения

Для реагирования и устранения последствий в результате возникновения этих ЧС требуются различные стратегии, методы и подходы в зависимости от сферы их возникновения [2 – 4]. В работе [2] рассматривается реагирование (массовая эвакуация) на природные ЧС. В работах [3 – 4] представлены подходы к эвакуации населения в случае экологических и техногенных ЧС. Рисунок 1 представляет укрупненную классификацию ЧС по сфере возникновения ЧС.

В основе методов реагирования на ЧС лежат различные методы моделирования потоков. В этих методах моделью эвакуационной сети является графовые и гиперграфовые представления транспортных сетей [5]. Форд и Фалкерсон [5] предложили метод определения максимального потока в динамической сети, который позволяет перемещать как можно больше пострадавших в убежища. Задача скорейшего прибытия, предложенная Гейлом [6], требует транспортировки максимального количества пострадавших, прибывающих в пункты назначения в каждый период времени.

Лексикографическая стратегия эвакуации [7] предполагает транспортировку потока в предписанном порядке, так что источники и/или стоки ранжируются. Идея «развернутой во времени», или «разнесенной во времени» сети Форда и Фалкерсона была расширена Чалметом и др. [8] для эвакуации из зданий. Ускоренная транспортировка эвакуируемых связана с проблемой потока наискорейшего прибытия [9]. В работе [9] указанная задача решена для эвакуации из больших зданий с помощью вероятностных сетей в динамических условиях, чтобы минимизировать общее время транспортировки людей [10]. В работе [11] потоковая задача решена как задача эвакуации, учитывающая различные приоритетные уровни для параметров сети. Задача потока минимальной стоимости была преобразована в задачу маршрутизации эвакуации Данном и Ньютоном [12], которые предложили алгоритмы для поиска кратчайших путей в сетях эвакуации, чтобы минимизировать общее расстояние перемещения. Миллер-Хукс и Паттерсон [13] расширили идею изменяющихся во времени параметров сетей, чтобы отправить максимальное количество пострадавших в соответствии с наискорейшей стратегией эвакуации в динамической сети с ограниченными пропускными способностями. На основании проведенного анализа литературных источников можно сделать вывод, что требуется расширение концепции лексикографической эвакуации, которая позволит определять приоритетность промежуточных решений и перемещать эвакуируемых в найденном порядке для максимизации числа спасаемых". Данный тип потоковой оптимизации актуален в реальных задачах эвакуации, так как он позволяет переместить большее число людей в безопасные зоны в связи с возможностью промежуточных узлов размещать дополнительный поток, что приводит к возможности перемещения людей из промежуточных вершин, не являющихся стоками.

Заметим, что характер исследуемой эвакуационной сети в проанализированных источниках часто неточен и не определен. Точные значения пропускной способности дуг сети и стоимости транспортировки неизвестны и могут быть только приблизительно оценены. Для решения этой проблемы предлагается использовать аппарат нечеткой логики и ее расширений [14 – 16]. В частности, были поставлены и решены различные потоковые задачи в нечетких условиях: нахождение максимального потока, потока минимальной стоимости, обобщенного потока, многопродуктового потока, потока наискорейшего прибытия как в статических, так и в динамических сетях [17]. Принятие решений (ПР) при эвакуации часто затруднено

рискованными и непрогнозируемыми аспектами эвакуации: некоторые детали эвакуации и данные, связанные со всем процессом, могут отсутствовать или иметь неопределенный характер. В этой связи были разработаны различные модели эвакуации с учетом неопределенности, а также изменяющейся во времени природы эвакуационных сетей [18 – 20]. Предложенные модели определяют методологическую основу данной работы.

Также существуют подходы к эвакуации людей с использованием мультиагентного моделирования и машинного обучения [21 – 22]. В работе [21] предложена программная реализация развитой мультиагентной модели эвакуации людей различных возрастных групп из помещений сложной формы. Для каждой возрастной группы было рассчитано и усреднено время полной эвакуации из помещения. Однако данная работа не учитывает нечеткий характер параметров сети, неточность и неполноту данных, а также не позволяет учесть различные критерии, влияющие на качество принимаемых решений. Кроме того, моделирование эвакуации осуществляется из помещений, что не позволяет применить данную модель к задачам массовой эвакуации при возникновении стихийных бедствий. В работе [22] предлагается объединение имитационных моделей эвакуации с методами машинного обучения для решения задач оценки эффективности управления эвакуации людей при пожарах в общественных зданиях. В работе представлена укрупненная структурная схема методики прогнозирования эффективности управления эвакуации людей с использованием методов машинного обучения. Однако в данной статье не представлены моделирование эвакуации, эксперименты, позволяющие сравнить эффективность разработанной модели с существующими. Кроме того, алгоритм, представленный структурной схемой, ориентирован на эвакуацию из помещений, не предназначен для решения задач массовой эвакуации людей на транспортных сетях, не включает блок принятия решения в нечетких условиях и не позволяет рассмотреть различные критерии эвакуации.

Зачастую принятие эвакуационных решений сопряжено со сложной стратегией реагирования, необходимостью учета различных факторов из разных областей, что требует экспертных знаний от разных лиц, принимающих решения (ЛПР) из разных областей [23 – 24]. В этой связи групповое принятие решений имеет наибольшее значение при эвакуационном реагировании. Кроме того, реагирование требует быстрых и эффективных решений о порядке, в котором будут эвакуированы люди из опасных зоны. Непрогнозируемые ЧС

происходят внезапно и влекут за собой трудности в определении наиболее безопасной стратегии эвакуации в убежища для транспортировки максимального количества эвакуируемых. В литературе по макроскопической эвакуации предлагается рассматривать критерий отдаленности от источника опасности как единственный критерий, влияющий на ранжирование эвакуационных стратегий [25]. Так как на приоритет того или иного убежища влияют и другие факторы, такие как вместимость помещения, которая задает верхний предел количества людей, находящихся в этом убежище, уровень безопасности помещения, близость к источнику опасности, сложность эвакуации и время на организацию эвакуации из этого места, то в работе предлагается учитывать эти критерии.

Лицам, принимающим решения об эвакуации, сложно дать точную оценку каждому критерию или указать численное значение при оценке опасных зон. Таким образом, процесс ПР требует включения нечетких рассуждений. Этот инструмент позволяет экспертам учитывать сомнения и уровень неопределенности при выборе конкретной оценки. В этом отношении теория нечетких множеств является полезным инструментом для оценки атрибутов в задачах многокритериального группового ПР (МКГПР). Нечеткие множества [14] изначально были предложены для представления неуверенности исследователя в виде функции принадлежности, имеющей уровни принадлежности относительно исходного множества. Развитие теории нечетких множеств привело к появлению различных обобщений нечетких множеств, призванных эффективнее обрабатывать возросший недостаток знаний и данных, сомнения экспертов, колебания при выборе функции принадлежности. Среди обобщений нечетких множеств выделяют нечеткие множества второго типа, нечеткие мультимножества, интуиционистские нечеткие множества, интуиционистские мягкие нечеткие множества, лингвистические аргументы, колеблющиеся нечеткие множества [26]. Нечеткое множество второго типа представляет принадлежность элемента множеству в виде нечёткого множества. Нечеткие мультимножества включают дублирующие элементы. Нечеткое колеблющееся множество (HFS) оперирует несколькими значениями степеней принадлежности [26]. Интуиционистское нечеткое множество состоит из функции принадлежности, функции непринадлежности и функции сомнения. МКГПР на основе интуиционистских нечетких множеств позволяет преобразовывать оценки лиц, принимающих решения, в нечеткие интуиционистские числа для эффективного моделирования. Именно наличие функции

сомнения обусловила выбор интуиционистских множеств для решения рассматриваемой в статье задачи.

В задачах ПР проблема определения весов атрибутов имеет важное значение. Подходы [27 – 28] оперируют весами атрибутов, заданными заранее, в то время как алгоритмы [29 – 30] предполагают, что веса атрибутов равны. Однако во время эвакуационного реагирования эти подходы не охватывают все аспекты входной информации, касающейся эвакуации: эксперты могут столкнуться с отсутствием знаний или недостаточной экспертизой по некоторым атрибутам. Более того, каждый эксперт имеет общий уровень надежности, который также необходимо учитывать. В связи с этим в этой статье веса экспертов отражают конкретные знания экспертов и не являются равными. В задачах реального ПР веса критериев не всегда точно известны. Если эксперты сомневаются и не имеют достаточной информации, они могут указать веса критериев в нечетком виде. Однако существующие подходы к нечёткому ПР не позволяют учесть нечёткие веса критериев. В этой связи предлагается оперировать нечёткими весами критериев в виде интуиционистских нечетких чисел.

С учетом рассмотренных подходов, их улучшений и модификаций предложен интегрированный алгоритм МГПР для выбора оптимальной стратегии при эвакуации. Данный алгоритм является многоуровневым, где на первом уровне осуществляется построение модели задачи в виде эвакуационной транспортной сети. На втором уровне производится ранжирование убежищ для нахождения их оптимального порядка на основе множества критериев. Определение приоритетности убежищ реализуется с помощью интуиционистского нечеткого гибридного оператора усреднения для учета неуверенности экспертов при оценке рассматриваемых критериев эвакуации. Предложенный гибридный оператор позволяет учесть веса критериев в виде нечетких интуиционистских значений на основе разработанных модифицированных операторов возведения нечетких чисел в нечеткую степень. На третьем уровне осуществляется макроскопическая потоковая динамическая эвакуация, позволяющая переместить максимальное количество людей из опасных зон.

Разработанный алгоритм опирается на теорию нечётких множеств, принятия решений в нечётких условиях и потоковую макроскопическую эвакуацию для эвакуации максимального числа людей за счет использования возможностей промежуточных вершин передавать дополнительный поток во все стоки в условиях неполноты

и неточности информации. Также разработанный алгоритм позволяет улучшить качество принимаемых решений в ситуациях, когда эксперты колеблются и сомневаются при оценке множества критериев эвакуации. К таким критериям были отнесены: вместимость помещения, которая задает верхний предел количества людей, находящихся в этом убежище, уровень безопасности помещения, близость к источнику опасности, сложность эвакуации и время на организацию эвакуации из этого места. Результаты алгоритма могут быть использованы как при эвакуации зданий, так и в задачах массовой эвакуации населения при масштабных ЧС.

2. Нечеткая логика, ее расширения и операторы агрегирования. Рассмотрим математический аппарат теории нечетких множеств, лежащий в основе разработанного метода принятия решений при эвакуации в нечеткой среде. Данный аппарат представляет собой обобщения нечётких множеств, а именно, интуиционистские нечёткие множества, позволяющие моделировать сомнения экспертов и их неуверенность при выборе конкретной степени принадлежности элемента нечёткому множеству более эффективно по сравнению с традиционными нечёткими множествами.

Нечеткая логика является эффективным методом объединения нескольких критериев с учетом опыта многих экспертов. Теория нечетких множеств, разработанная Л. Заде в 1965 году [31], стала мощным инструментом в различных областях современного общества. Среди них – система управления искусственным кардиостимулятором, химические реакторы и устройства, система контроля кровяного давления, система оповещения о сердечных заболеваниях, энергосистема, система управление дорожным движением, биомедицина [32]. Нечеткое множество расширяет характеристическую функцию, принимающую значение 0 или 1, до функции принадлежности, которая может принимать любое значение из закрытого интервала $[0,1]$. Рассмотрим концепцию нечеткого множества.

Определение 1. Пусть X – непустое множество. Нечеткое множество на множестве X – это множество F , такое что:

$$F = \{(x, \mu_F(x)); x \in X\}, \quad (1)$$

где $\mu_F: X \rightarrow [0,1]$ называется функцией принадлежности F . $\mu_A(x)$ показывает степень принадлежности элемента x множеству F и принимает значение из интервала $[0,1]$.

Однако функция принадлежности является только однозначной функцией, которая не может использоваться для выражения поддержки и возражения одновременно во многих практических ситуациях.

В задачах ПР, моделирования, управления системами эксперты могут не обладать точным или достаточным уровнем знаний о проблемной области из-за сложности социально-экономической среды. В таких случаях обычно присутствует некоторая неопределенность в предпочтениях экспертов относительно рассматриваемых объектов, тем самым демонстрируются характеристики утверждения, отрицания и колебания. Например, при голосовании, помимо «поддержки» и «возражения», обычно есть «воздержание», что указывает на колебание и неопределенность избирателя относительно объекта. Поскольку нечеткое множество не может быть использовано для полного выражения всей информации в таких проблемах, его применимость часто ограничена во многих практических ситуациях. Для таких ситуаций могут использованы обобщения нечетких множеств, в частности, интуиционистские нечёткие множества. Это актуально и для задач ПР при эвакуационном моделировании, когда эксперт или группа экспертов не могут предоставить оценку эвакуационной альтернативы или критерия в виде конкретной степени принадлежности, а используют помимо степени принадлежности элемента нечеткому множеству также степень непринадлежности и степень сомнения. Рассмотрим аппарат теории интуиционистских нечётких множеств, позволяющий эффективно принимать решения при эвакуационном моделировании и реагировании в нечетких условиях и условиях неполноты информации, более подробно.

Атанассов в 1986 году [33] обобщил теорию нечетких множеств с помощью концепции интуиционистского нечеткого множества (IFS), которая включала степень сомнения (колебания) или функцию неопределенности, а также функцию принадлежности и функцию непринадлежности элемента множеству IFS. Интуиционистский индекс не обязательно равен нулю, как в нечетких множествах. Использование IFS обеспечивает всестороннее и целостное описание нечетких параметров, что обеспечивает получение эффективных результатов при исходной неопределенности.

Последние несколько десятилетий исследователи уделяли большое внимание изучению теории IFS и достигли значимых результатов [34]. Атанассов [33] предложил несколько основных операций над IFS, включая «пересечение», «объединение»,

«дополнение» и «возведение в степень». Обзор основных операторов агрегирования, лежащих в основе МГПР, представлен в работе [35]. Однако, поскольку изучение теории IFS расширяется как в глубину, так и в ширину, эффективные агрегирование и обработка интуиционистской нечеткой информации становятся необходимыми и все более важными.

Рассмотрим основные концепции, лежащие в основе теории IFS.

Определение 2. Пусть X – непустое множество. IFS на множестве X – это множество $\tilde{A}$ такое, что:

$$\tilde{A} = \{(x, \mu_A(x), \nu_A(x)); x \in X\}, \quad (2)$$

где $\mu_A(x): X \rightarrow [0,1]$ и $\nu_A(x): X \rightarrow [0,1]$ удовлетворяют условиям $0 \leq \mu_A(x) \leq 1$ для каждого $x \in X$.

$\pi_A(x)$ в $\tilde{A}$ называется степенью нерешительности, или пределом колебания, или функцией неопределенности, которая измеряет уровень нерешительности:

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x). \quad (3)$$

Интуиционистским нечетким числом (IFN) [36] называется число $\alpha = (\mu_\alpha, \nu_\alpha)$, где $\mu_\alpha \in [0,1]$, $\nu_\alpha \in [0,1]$, $\mu_\alpha + \nu_\alpha \leq 1$.

Для сравнения IFN используется функция оценки $s(\alpha)$:

$$s(\alpha) = \mu_\alpha - \nu_\alpha, s(\alpha) \in [-1,1]. \quad (4)$$

Если IFN равны, их можно сравнить с помощью функции точности h [36]:

$$h(\alpha) = \mu_\alpha + \nu_\alpha, \quad (5)$$

где $h(\alpha)$ – степень точности $h(\alpha) = \mu_\alpha$. Чем больше $h(\alpha)$, тем выше степень точности IFN.

Определение 3. Пусть α_1 и α_2 два IFN; $s(\alpha_1)$ и $s(\alpha_2)$ – функции оценки этих значений; $h(\alpha_1)$ и $h(\alpha_2)$ – функции точности α , тогда:

- 1) Если $s(\alpha_1) < s(\alpha_2)$, то $\alpha_1 < \alpha_2$;
- 2) Если $s(\alpha_1) = s(\alpha_2)$, то
 - 2.1) если $h(\alpha_1) < h(\alpha_2)$, то $\alpha_1 > \alpha_2$;
 - 2.2) если $h(\alpha_1) > h(\alpha_2)$, то $\alpha_1 > \alpha_2$;
 - 2.3) если $h(\alpha_1) = h(\alpha_2)$, то $\alpha_1 = \alpha_2$ [36].

Пусть $\tilde{A} = (\mu_{\tilde{A}}, \nu_{\tilde{A}})$ и $\tilde{B} = (\mu_{\tilde{B}}, \nu_{\tilde{B}})$ – два IFN на множестве X ; тогда базовые операции с IFN определяются следующим образом:

$$\tilde{A} \oplus \tilde{B} = (\mu_{\tilde{A}}(x) + \mu_{\tilde{B}}(x) - \mu_{\tilde{A}}(x)\mu_{\tilde{B}}(x), \nu_{\tilde{A}}(x) \times \nu_{\tilde{B}}(x), x \in X). \quad (6)$$

$$k\tilde{A} = (1 - (1 - \mu_{\tilde{A}}(x))^k, \nu_{\tilde{A}}(x)^k, x \in X, k > 0. \quad (7)$$

$$\alpha^\lambda = (\mu_\alpha^\lambda, 1 - (1 - \nu_\alpha)^\lambda), \lambda > 0. \quad (8)$$

Процесс принятия решения в задачах эвакуации для выбора наилучших эвакуационных альтернатив требует агрегации всех входных данных в одно значение с помощью операторов агрегирования. До настоящего времени было предложено много операторов для агрегирования информации в различных средах принятия решений [37].

Наиболее распространенными операторами для агрегирования аргументов являются оператор взвешенного усреднения (WA), взвешенный геометрический оператор (WG), оператор упорядоченного взвешенного усреднения (OWA) и упорядоченный взвешенный геометрический оператор (OWG) [26, 36].

Первоначально данные операторы использовались для агрегации аргументов, принимающих значения из множества действительных чисел. Позже они были расширены для адаптации к неопределенным и нечетким лингвистическим средам [38 – 39].

Рассмотрим существующие операторы агрегирования:

Определение 4. Пусть $\alpha_j = (\mu_{\alpha_j}, \nu_{\alpha_j})$, $j=1, 2, \dots, n$ представляет собой коллекцию IFNs, тогда:

$$IFWA_\omega(\alpha_1, \alpha_2, \dots, \alpha_n) = \omega_1 \alpha_1 \oplus \omega_2 \alpha_2 \oplus \dots \oplus \omega_n \alpha_n, \quad (9)$$

где $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$ – весовой вектор, связанный с оператором α_j , $j=1, 2, \dots, n$, $\omega_j \in [0, 1]$ $j=1, 2, \dots, n$ и $\sum_{j=1}^n \omega_j = 1$.

Определение 5. Агрегированное значение, полученное с помощью оператора IFWA представляет собой

$$IFWA_\omega(\alpha_1, \alpha_2, \dots, \alpha_n) = (1 - \prod_{j=1}^n (1 - \mu_{\alpha_j})^{\omega_j}, \prod_{j=1}^n (\nu_{\alpha_j})^{\omega_j}). \quad (10)$$

Определение 6. Пусть $\alpha_j = (\mu_{\alpha_j}, \nu_{\alpha_j})$, $j=1,2,\dots,n$ представляет собой коллекцию IFNs, тогда

$$IFWG_{\omega}(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha_1^{\omega_1} \otimes \alpha_2^{\omega_2} \otimes \dots \otimes \alpha_n^{\omega_n}. \quad (11)$$

IFWG называется интуиционистским нечётким взвешенным геометрическим оператором (IFWG), где $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$ – вектор весов α_j , $j=(1,2,\dots,n)$, $\omega_j \in [0,1]$ $j=(1,2,\dots,n)$ и $\sum_{j=1}^n \omega_j = 1$.

Определение 7. Агрегированное значение, полученное с помощью оператора IFWG, представляет собой

$$IFWG_{\omega}(\alpha_1, \alpha_2, \dots, \alpha_n) = \left(\prod_{j=1}^n \mu_{\alpha_j}^{\omega_j}, 1 - \prod_{j=1}^n (1 - \nu_{\alpha_j})^{\omega_j} \right), \quad (12)$$

где $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$ – вектор весов α_j , $j=(1,2,\dots,n)$, $\omega_j \in [0,1]$ $j=(1,2,\dots,n)$ и $\sum_{j=1}^n \omega_j = 1$.

Оба оператора IFWA и IFWG сначала взвешивают все аргументы, а затем агрегируют эти взвешенные аргументы. Однако оператор IFWG гораздо более чувствителен к исходным аргументам.

Определение 8. Интуиционистский нечеткий упорядоченный оператор взвешенного усреднения (IFOWA) – это такое отображение, что $= (w_1, w_2, \dots, w_n)^T$, $w_j \in [0,1]$ $j=(1,2,\dots,n)$ and $\sum_{j=1}^n w_j = 1$.

$$IFOWA_w(\alpha_1, \alpha_2, \dots, \alpha_n) = w_1 \alpha_{\sigma(1)} \oplus w_2 \alpha_{\sigma(2)} \oplus \dots \oplus w_n \alpha_{\sigma(n)}, \quad (13)$$

где $(\sigma(1), \sigma(2), \dots, \sigma(n))$ является перестановкой $(1,2,\dots,n)$, такой что $\alpha_{\sigma(j-1)} \geq \alpha_{\sigma(j)}$ для j .

Определение 9. Агрегированное значение на основе оператора IFOWA – это интуиционистские нечёткие значения (IFV) в форме

$$IFOWA_w(\alpha_1, \alpha_2, \dots, \alpha_n) = \left(1 - \prod_{j=1}^n (1 - \mu_{\alpha_{\sigma(j)}})^{w_j}, \prod_{j=1}^n \nu_{\alpha_{\sigma(j)}}^{w_j} \right). \quad (14)$$

$w = (w_1, w_2, \dots, w_n)^T$ – вектор весов, связанный с оператором IFOWA, $w_j \in [0,1]$ $j=(1,2,\dots,n)$ и $\sum_{j=1}^n w_j = 1$.

Определение 10. Интуиционистский нечеткий упорядоченный взвешенный геометрический оператор (IFOWA) – это такое

отображение $w = (w_1, w_2, \dots, w_n)^T$, что $w_j \in [0,1]$ $j=(1,2,\dots,n)$ и $\sum_{j=1}^n w_j = 1$.

$$IFOWG_w(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha_{\sigma(1)}^{w_1} \otimes \alpha_{\sigma(2)}^{w_2} \otimes \dots \otimes \alpha_{\sigma(n)}^{w_n}, \quad (15)$$

где $(\sigma(1), \sigma(2), \dots, \sigma(n))$ – перестановка $(1,2,\dots,n)$, такая что $\alpha_{\sigma(j-1)} \geq \alpha_{\sigma(j)}$ для любых j .

Определение 11. Агрегированное значение с использованием оператора IFOWG – это IFN в форме

$$IFOWG_w(\alpha_1, \alpha_2, \dots, \alpha_n) = \left(\prod_{j=1}^n \mu_{\alpha_{\sigma(j)}}^{w_j}, 1 - \prod_{j=1}^n (1 - \nu_{\alpha_{\sigma(j)}})^{w_j} \right). \quad (16)$$

$w = (w_1, w_2, \dots, w_n)^T$ – это весовой вектор, связанный с оператором IFOWG, $w_j \in [0,1]$ $j=(1,2,\dots,n)$ и $\sum_{j=1}^n w_j = 1$.

Операторы IFOWA и IFOWG сначала переставляют заданные аргументы в порядке убывания, а затем взвешивают упорядоченные позиции этих аргументов. Упорядоченные взвешенные операторы агрегируют упорядоченные аргументы и веса их упорядоченных позиций, поэтому аргумент не связан с определенным весом. Вместо этого определенный вес связан с определенной упорядоченной позицией аргументов.

Вышеупомянутые операторы IFWA и IFWG назначают веса заданным IFN, тогда как операторы IFOWA и IFOWG назначают веса упорядоченным позициям указанных чисел. Однако каждый из операторов рассматривает один аспект. Чтобы объединить эти два аспекта, были предложены интуиционистские нечёткие операторы гибридного усреднения (IFHA) [40], которые взвешивают предопределенные IFN и их упорядоченные позиции.

Определение 12. Оператор IFHA представляет собой такое отображение, что

$$IFHA_{\omega,w}(\alpha_1, \alpha_2, \dots, \alpha_n) = w_1 \hat{\alpha}_{\sigma(1)} \oplus w_2 \hat{\alpha}_{\sigma(2)} \oplus \dots \oplus w_n \hat{\alpha}_{\sigma(n)}, \quad (17)$$

где $w = (w_1, w_2, \dots, w_n)^T$ – вектор весов, связанный с оператором IFHA, $w_j \in [0,1]$, $j=(1,2,\dots,n)$ и $\sum_{j=1}^n w_j = 1$. $\hat{\alpha}_j = n\omega_j\alpha_j$, $j=1,2,\dots,n$, $\hat{\alpha}_{\sigma(1)}, \hat{\alpha}_{\sigma(2)}, \dots, \hat{\alpha}_{\sigma(n)}$ – перестановка взвешенных IFNs, таких что $(\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_n)$, $\hat{\alpha}_{\sigma(j)} \geq \hat{\alpha}_{\sigma(j+1)}$, $(j=1,2,\dots,n-1)$; $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$ –

вектор весов α_j , $j=(1,2,...,n)$, $\omega_j \in [0,1]$ $j=(1,2,...,n)$ и $\sum_{j=1}^n \omega_j = 1$. Коэффициент балансировки равен n .

Определение 13. Пусть $\hat{\alpha}_{\sigma(j)} = (\mu_{\hat{\alpha}_{\sigma(j)}}, \nu_{\hat{\alpha}_{\sigma(j)}})$, $(j=1,2,...,n)$.

Агрегированное значение с использованием оператора IFHA – это IFN в форме:

$$IFHA_{\omega,w}(\alpha_1, \alpha_2, \dots, \alpha_n) = (1 - \prod_{j=1}^n (1 - \mu_{\hat{\alpha}_{\sigma(j)}})^{\omega_j} \prod_{j=1}^n \nu_{\hat{\alpha}_{\sigma(j)}}^{\omega_j}). \quad (18)$$

Определение 14. Интуиционистский нечеткий гибридный геометрический оператор (IFHG) – это такое отображение, что

$$IFHG_{\omega,w}(\alpha_1, \alpha_2, \dots, \alpha_n) = \hat{\alpha}_{\sigma(1)}^{w_1} \otimes \hat{\alpha}_{\sigma(2)}^{w_2} \otimes \dots \otimes \hat{\alpha}_{\sigma(n)}^{w_n}, \quad (19)$$

где $\hat{\alpha}_j = \alpha_j^{n\omega_j}$, $j=(1,2,...,n)$, $(\hat{\alpha}_{\sigma(1)}, \hat{\alpha}_{\sigma(2)}, \dots, \hat{\alpha}_{\sigma(n)})$ – перестановка взвешенных IFNs, таких что $(\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_n)$, $\hat{\alpha}_{\sigma(1)} \geq \hat{\alpha}_{\sigma(j+1)}$, $(j=1,2,...,n-1)$;

Определение 15. Пусть $\hat{\alpha}_{\sigma(j)} = (\mu_{\hat{\alpha}_{\sigma(j)}}, \nu_{\hat{\alpha}_{\sigma(j)}})$, $(j=1,2,...,n)$.

Агрегированное значение, полученное с помощью оператора IFHG – это IFN в виде

$$IFHG_{\omega,w}(\alpha_1, \alpha_2, \dots, \alpha_n) = \left(\prod_{j=1}^n \mu_{\hat{\alpha}_{\sigma(j)}}^{\omega_j}, 1 - \prod_{j=1}^n (1 - \nu_{\hat{\alpha}_{\sigma(j)}})^{\omega_j} \right). \quad (20)$$

Гибридные операторы IFHA и IFHG требуют выполнения следующих шагов:

1. Взвесить IFNs $\alpha_j, j=(1,2,...,n)$ в соответствии со значениями связанных весов $\omega_j, j=(1,2,...,n)$ и найти $\omega_j \alpha_j$ или $\alpha_j^{\omega_j} j=(1,2,...,n)$. Затем умножить полученные значения на балансировочный коэффициент n , чтобы найти взвешенные IFNs $n\omega_j \alpha_j$ или $\alpha_j^{n\omega_j} j=(1,2,...,n)$.

2. Переупорядочить взвешенные IFNs $\hat{\alpha} = n\omega_j \alpha_j$ или $\hat{\alpha} = \alpha_j^{n\omega_j}, j=(1,2,...,n)$ в порядке убывания, $(\hat{\alpha}_{\sigma(1)}, \hat{\alpha}_{\sigma(2)}, \dots, \hat{\alpha}_{\sigma(n)})$ или $(\hat{\alpha}_{\sigma(1)}, \hat{\alpha}_{\sigma(2)}, \dots, \hat{\alpha}_{\sigma(n)})$, где $\hat{\alpha}_{\sigma(j)}$ – это j -ое наибольшее значение $n\omega_i \alpha_i, i=(1,2,...,n)$, и $\hat{\alpha}_{\sigma(j)}$ – это j -ое наибольшее значение $\alpha_j^{n\omega_j} j=(1,2,...,n)$.

3. Взвесить полученные упорядоченные веса IFNs $\hat{\alpha}_{\sigma(j)}$ или $\hat{\alpha}_{\sigma(j)}^w$ $j=(1,2,...,n)$ с использованием весов IFHA или IFHG w_j $j=(1,2,...,n)$. Наконец, агрегировать все взвешенные IFNs $w_j\hat{\alpha}_{\sigma(j)}$ или $\hat{\alpha}_{\sigma(j)}^w$ $j=(1,2,...,n)$ в коллективное значение.

Рассмотрев и проанализировав существующие операторы агрегирования, заметим, что в задачах реального принятия решений веса критериев не всегда представляются в четком виде. Если эксперты сомневаются при выборе весовых коэффициентов, то веса критериев следует задавать в виде интуиционистских нечетких чисел. Традиционные операторы агрегирования не позволяют управлять нечеткими весами критериев. В связи с этим в работе предлагается методика, позволяющая оперировать нечеткими весами критериев и приводящая к эффективному ПР в случае сомнений и колебаний экспертов, их неуверенности при выборе степеней принадлежности.

3. Модифицированные операторы агрегирования с нечеткими весами критериев. Введем модифицированные операторы агрегирования, позволяющие управлять нечеткими весами критериев, лежащие в основе интуиционистского нечеткого гибридного оператора усреднения. Данный оператор будет использован для ранжирования эвакуационных альтернатив, представляющий собой помещения для эвакуации с целью максимизации числа спасенных людей.

Пусть $\alpha = (\mu_r, \nu_r)$, $\alpha_1 = (\mu_{r_1}, \nu_{r_1})$ и $\alpha_2 = (\mu_{r_2}, \nu_{r_2})$ – три IFVs, $k > 0$. Тогда справедливы следующие законы [36]:

- (1) $\alpha_1 \oplus \alpha_2 = (\mu_{r_1} + \mu_{r_2} - \mu_{r_1}\mu_{r_2}, \nu_{r_1}\nu_{r_2})$;
- (2) $\alpha_1 \otimes \alpha_2 = (\mu_{r_1}\mu_{r_2}, \nu_{r_1} + \nu_{r_2} - \nu_{r_1}\nu_{r_2})$;
- (3) $k\alpha = (1 - (1 - \mu_r)^k, \nu_r^k)$;
- (4) $\alpha^k = (\mu_r^k, 1 - (1 - \nu_r)^k)$.

Пусть $\alpha = (\mu_r, \nu_r)$ и $\tau = (\lambda, \theta)$ – два IFVs. Учитывая операционный закон (3) и равенство $IFWA_\omega(\alpha_1, \alpha_2, \dots, \alpha_n) = \omega_1\alpha_1 \oplus \omega_2\alpha_2 \oplus \dots \oplus \omega_n\alpha_n$, получаем

$$\tau\alpha = (1 - (1 - \mu_r)^\lambda, \nu_r^{1-\theta}). \quad (21)$$

Учитывая операционный закон (4) и равенство $IFWG_\omega(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha_1^{\omega_1} \otimes \alpha_2^{\omega_2} \otimes \dots \otimes \alpha_n^{\omega_n}$, получаем

$$\alpha^\tau = (\mu_r^{1-\theta}, 1 - (1 - \nu_r)^\lambda). \quad (22)$$

Таким образом, модифицированные операторы агрегирования с интуиционистскими нечеткими весами критериев представляются в виде:

$$IFWA_{\tau}(\alpha_1, \alpha_2, \dots, \alpha_n) = \tau_1 \alpha_1 \oplus \tau_2 \alpha_2 \oplus \dots \oplus \tau_n \alpha_n = \left(1 - \prod_{j=1}^n (1 - \mu_{\alpha_j})^{\lambda_j}, \prod_{j=1}^n (v_{\alpha_j})^{1-\theta_j}\right). \quad (23)$$

$$IFWG_{\tau}(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha_1^{\tau_1} \otimes \alpha_2^{\tau_2} \otimes \dots \otimes \alpha_n^{\tau_n} = \left(\prod_{j=1}^n \mu_{\alpha_j}^{1-\theta_j}, 1 - \prod_{j=1}^n (1 - v_{\alpha_j})^{\lambda_j}\right). \quad (24)$$

$$IFOWA_w(\alpha_1, \alpha_2, \dots, \alpha_n) = \tau_1 \alpha_{\sigma(1)} \oplus \tau_2 \alpha_{\sigma(2)} \oplus \dots \oplus \tau_n \alpha_{\sigma(n)} = \left(1 - \prod_{j=1}^n (1 - \mu_{\alpha_{\sigma(j)}})^{\lambda_j}, \prod_{j=1}^n v_{\alpha_{\sigma(j)}}^{1-\theta_j}\right). \quad (25)$$

$$IFOWG_w(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha_{\sigma(1)}^{\tau_1} \otimes \alpha_{\sigma(2)}^{\tau_2} \otimes \dots \otimes \alpha_{\sigma(n)}^{\tau_n} = \left(\prod_{j=1}^n \mu_{\alpha_{\sigma(j)}}^{1-\theta_j}, 1 - \prod_{j=1}^n (1 - v_{\alpha_{\sigma(j)}})^{\lambda_j}\right). \quad (26)$$

С учетом вышеизложенного модифицированные гибридные операторы представляются следующим образом: IFHA и IFHG.

Согласно операционному закону (3) и равенству $IFHA_{\omega,w}(\alpha_1, \alpha_2, \dots, \alpha_n) = w_1 \hat{\alpha}_{\sigma(1)} \oplus w_2 \hat{\alpha}_{\sigma(2)} \oplus \dots \oplus w_n \hat{\alpha}_{\sigma(n)}$, получаем

$$IFHA_{\tau,w} = w_1 \hat{\alpha}_{\sigma(1)} \oplus w_2 \hat{\alpha}_{\sigma(2)} \oplus \dots \oplus w_n \hat{\alpha}_{\sigma(n)} = \left(1 - \prod_{j=1}^n (1 - \mu_{\hat{\alpha}_{\sigma(j)}})^{w_j} \prod_{j=1}^n v_{\hat{\alpha}_{\sigma(j)}}^{w_j}\right), \text{ где } \hat{\alpha}_j = n \tau_j \alpha_j = \left(1 - \prod_{j=1}^n (1 - \mu_{\alpha_j})^{n \lambda_j}, \prod_{j=1}^n (v_{\alpha_j})^{n(1-\theta_j)}\right). \quad (27)$$

Согласно операционному закону (4) и равенству $IFHG_{\omega,w}(\alpha_1, \alpha_2, \dots, \alpha_n) = \hat{\alpha}_{\sigma(1)}^{w_1} \otimes \hat{\alpha}_{\sigma(2)}^{w_2} \otimes \dots \otimes \hat{\alpha}_{\sigma(n)}^{w_n}$, получаем

$$IFHG_{\tau,w} = \hat{\alpha}_{\sigma(1)}^{w_1} \otimes \hat{\alpha}_{\sigma(2)}^{w_2} \otimes \dots \otimes \hat{\alpha}_{\sigma(n)}^{w_n} = \left(\prod_{j=1}^n \mu_{\hat{\alpha}_{\sigma(j)}}^{w_j}, 1 - \prod_{j=1}^n (1 - v_{\hat{\alpha}_{\sigma(j)}})^{w_j}\right), \text{ where } \hat{\alpha}_j = \alpha_j^{n \omega_j} = \left(\prod_{j=1}^n \mu_{\alpha_j}^{n(1-\theta_j)}, 1 - \prod_{j=1}^n (1 - v_{\alpha_j})^{n \lambda_j}\right). \quad (28)$$

Построенные модифицированные нечеткие операторы позволяют работать с интуиционистскими нечеткими весами критериев и лежат в основе интуиционистского нечеткого гибридного оператора усреднения, которые эффективно применяется для многокритериального группового принятия решений по выбору оптимальной стратегии эвакуации. Под стратегией эвакуации понимается порядок расположения помещений для эвакуации в эвакуационной сети, которая представлена транспортной сетью.

4. Постановка задачи динамической макроскопической эвакуации. Представим основные определения, лежащие в основе макроскопической эвакуации людей при эвакуации. Под макроскопической эвакуацией понимается эвакуации людей как потока без учета индивидуальных характеристик отдельного человека. При макроскопической эвакуации моделью эвакуационной сети является графовые и гиперграфовые представления транспортных сетей [5, 7, 17 – 19]. Эвакуационная сеть определяется как транспортная сеть, соответствующая территории (или зданию), подлежащим эвакуации, где пересечения дорог (или помещения в здании) представляют собой узлы сети (вершины), а связи между ними (дороги на эвакуационной территории или коридоры) обозначаются рёбрами (дугами). В такой сети есть источники – области, подвергшиеся опасности, из которых осуществляется транспортировка, и стоки – убежища, представляющие собой безопасные области или помещения, куда перемещают людей. Все остальные узлы таких сетей называют промежуточными (пересечения дорог или помещения). Изначально эвакуируемые находятся в вершинах-источниках, а впоследствии осуществляется транспортировка в безопасные зоны – вершины-стоки. Вершины и рёбра эвакуационной сети ограничены пропускными способностями, отражающими максимальную величину эвакуационного потока, который может проходить по этому ребру или размещаться в узле. Группа эвакуируемых, проходящая через сеть с течением времени, моделируется как поток. План эвакуации зависит от количества источников, стоков, параметров, заданных на дугах и узлах, таких как постоянные, зависящие от времени или зависящие от потока пропускные способности или параметры времени прохождения, а также дополнительных ограничений. Алгоритм является универсальным и позволяет моделировать изменения в пропускных способностях в зависимости от различных факторов. Например, если в здании установлены газоанализаторы, показывающие уровень задымления, пропускная способность будет снижаться. Если

рассматривается транспортная сеть, то пропускные способности могут изменяться в зависимости от времени отправления потока эвакуируемых с учетом часов пик и пробок. Кроме того, уменьшение пропускных способностей может быть связано с обрушением деревьев на дороге, обвалом зданий другими препятствиями, возникающими на конкретном эвакуационном пути.

Приведем постановку задачи эвакуации с учетом предложенной нечеткой логики.

Определение 16. Нечеткая динамическая сеть $\tilde{G} = (X, \tilde{A})$ представляет собой сеть, включающую два множества: множество узлов $X = \{x_1, x_2, \dots, x_n\}$ и множество дуг $\tilde{A} = \{ \langle \mu_A < x_i, x_j \rangle / \langle x_i, x_j \rangle \rangle, \langle x_i, x_j \rangle \in X^2 \}$ с определенной функцией принадлежности $\mu_A(x_i, x_j)$ [17].

Дана нечеткая эвакуационная динамическая сеть в виде ориентированного графа $\tilde{G} = (X, \tilde{A}, \tilde{u}(a), \tilde{u}(x), \tau, S, I, D, T)$, где X – множество узлов, $|X| = n$; $\tilde{A}$ – множество нечетких дуг, $|\tilde{A}| = m$; $\tilde{u}(a)$ обозначает нечеткую пропускную способность дуги $\tilde{a} \in \tilde{A}$. Нечеткая пропускная способность $\tilde{u}(a)$ для всех $\tilde{a} = (x_i, x_j) \in A$ указывает максимальный объем потока, который может войти в дугу a из ее хвоста x_i за период времени. Следовательно, если поток стартует из x_i в момент времени θ , чтобы двигаться в x_j , он достигает x_j в момент времени $\theta + \tau(a)$. $\tilde{u}(x)$ обозначает пропускную способность узла x , которая ограничивает объем потока, разрешенный для хранения в x ; $\tau(a)$ – время прохождения дуги $\tilde{a} \in \tilde{A}$, необходимое потоку для прохождения от хвоста до головы узла; S – множество источников; I – множество промежуточных узлов (все вершины сети кроме источников и стоков); D – множество стоков, T – заданный параметр, отражающий возможный период времени, в течение которого должна быть завершена эвакуация. T делится на временные шаги $T = \{0, 1, \dots, T\}$. Для $\tilde{a} \in \tilde{A}$, $\tilde{u}(a) \geq 0$ и $\forall x \in X$, $\tilde{u}(x) \geq 0$.

Для дуги $\tilde{a} = (x_i, x_j) \in A$ узел $x_i \in X$ известен как хвостовой узел $tail(a) = x_i$. $x_j \in X$ рассматривается как головной узел $head(a) = x_j$. Множество дуг, исходящих из узла x_i , равно $L(x_i) = \{(x_i, x_j) \in \tilde{A}, x_i \in X\}$, а множество входящих дуг в узел x_i равно $E(x_i) = \{(x_i, x_j) \in \tilde{A}, x_i \in X\}$.

Определение 17. Пусть $\tilde{G} = (X, \tilde{A}, \tilde{u}(a), \tilde{u}(x), \tau, S, I, D, T)$ – исходная динамическая сеть. «Расширенная во времени» копия исходной динамической сети – это статическая сеть $\tilde{G}^* = (X^*, \tilde{A}^*)$, в которой $X^* = \{x_i(\theta) : x_i \in X, \theta = 0, 1, \dots, T\}$. Если поток может быть

сохранен в узлах, предполагаем, что существуют промежуточные дуги удержания $\tilde{A}^H = \{(x_i(\theta), x_i(\theta + 1)), x_i \in X, \theta = 0, 1, \dots, T\}$. Таким образом, множество дуг $\tilde{A}^*$ статической развернутой во времени сети $A^* = \tilde{A}^H \cup \tilde{A}^M$, где $\tilde{A}^M$ является множеством дуг движения $\tilde{A}^M = \{(x_i(\theta), x_i(\theta + \tau(x_i, x_j))), (x_i, x_j) \in X, \theta = 0, 1, \dots, T\}$.

Приведем модифицированную постановку задачи с учетом промежуточного хранения. Пусть $\tilde{G} = (X, \tilde{A}, \tilde{u}(a), \tilde{u}(x), \tau, S, I, D, T)$ – динамическая эвакуационная сеть [41 – 42] учитывающая возможность промежуточных вершин сохранять и перемещать эвакуационный поток в убежища (промежуточное хранение), $\tilde{v}$ – динамический поток $S \rightarrow D$, такой что $A \times T \rightarrow \mathbb{R}^+$, и $\tilde{v}(x_i, \theta), I \times T \rightarrow \mathbb{Z}^+$ – объем потока, который хранится в промежуточных узлах в течение времени $\theta \in T$. Таким образом, динамическая модель потока с промежуточным хранением задается в уравнениях (29)-(32):

$$val(\tilde{v}, T) = \sum_{a \in L(S)} \sum_{\vartheta=0}^T \tilde{v}(a, \vartheta) = \sum_{a \in E(D)} \sum_{\vartheta=\tau(a)}^T \tilde{v}(a, \vartheta - \tau(a)) + \sum_{x_i \in I, u(x_i) > 0} \tilde{v}(x_i, T) \rightarrow \max, \quad (29)$$

$$\begin{aligned} val(\tilde{v}, \theta) &= \sum_{a \in E(x_i)} \sum_{\vartheta=\tau(a)}^T \tilde{v}(a, \vartheta - \tau(a)) \\ &- \sum_{a \in L(x_i)} \sum_{\vartheta=0}^T \tilde{v}(a, \vartheta) \geq 0, \forall x_i \in I, \theta \in T. \end{aligned} \quad (30)$$

$$0 \leq \tilde{v}(a, \theta) \leq \tilde{u}(a, \theta), \forall a \in \tilde{A}, \theta \in T, \quad (31)$$

$$0 \leq \tilde{v}(x_i, \theta) \leq \tilde{u}(x_i), \forall x_i \in I, \theta \in T. \quad (32)$$

где $val(\tilde{v})$ максимизирует целевую функцию (29) относительно ограничений (30-32). Уравнение (30) иллюстрирует, что общий объем эвакуационного потока из источников не обязательно достигает стоков, но может быть размещен в промежуточных узлах без нарушения ограничений пропускной способности узлов для дальнейшего перемещения в безопасные зоны. Уравнения (31-32) показывают верхние пределы эвакуационного потока вдоль дуги и в промежуточных узлах.

В работе для максимизации числа спасенных жизней предлагается транспортировать эвакуационный поток сначала в убежища-стоки, после чего осуществлять транспортировку людей из промежуточных вершин-помещений с учетом ограниченных пропускных способностей. Разработанный подход реализует перемещение максимального числа эвакуируемых из опасных зон как сумму числа эвакуируемых в безопасные зоны и избыточного эвакуационного потока, который не достигает пунктов назначения. Избыточный поток сохраняется в промежуточных узлах в течение заданного времени, а затем перемещается в безопасные зоны. При этом порядок перемещения потока из промежуточных узлов в безопасные зоны определяется с помощью многокритериального группового принятия решений на основе интуиционистского нечеткого гибридного оператора усреднения.

5. Интегрированный нечеткий алгоритм принятия решений при эвакуации. Процесс принятия многокритериальных решений на основе оператора агрегации IFNA в работе представляется следующим образом. Задан набор, состоящий из n эвакуационных альтернатив $A = \{A_1, A_2, \dots, A_n\}$ (узлы, из которых необходимо эвакуировать людей) и m атрибутов $X = \{X_1, X_2, \dots, X_m\}$ (критерии, влияющие на эвакуацию: вместимость помещения, легкость эвакуации, уровень безопасности помещения, время организации эвакуации из этого места и отдаленность от источника опасности.). Пусть $\omega = (\omega_1, \omega_2, \dots, \omega_m)^T$ – весовой вектор атрибутов, $\omega_j \in [0,1]$ $j=(1,2,\dots,n)$ и $\sum_{j=1}^n \omega_j = 1$. Группа экспертов дает оценки альтернативам с помощью интуиционистских значений. Каждое значение IFS A_i задается уравнением (2) и указывает на то, что эксперты сомневаются и имеют некоторый уровень колебаний. В уравнении (2), $\mu_{A_i}(x_j)$ – функция принадлежности, которая указывает на уровень уверенности эксперта относительно критериев x_j ; $\nu_{A_i}(x_j)$ – функция непринадлежности, которая показывает, в какой степени альтернатива A_i не соответствует критериям x_j . Существует индекс неопределенности, который измеряет уровень колебаний по уравнению (3). Далее обрабатывается интуиционистская нечеткая матрица принятия решений, элементы которой $r'_{ij} = (r_m, r_n)$ указывают, что альтернатива A_i удовлетворяет критериям x_i (r_m); и наоборот, r_n показывает, что альтернатива не удовлетворяет критериям x_i .

Для реализации концепций интуиционистских нечётких множеств, принятия решений и нечетких интуиционистских операторов агрегирования и макроскопической эвакуации в

транспортной сети разработан интегрированный алгоритм, состоящий из трех уровней, структурная схема которого представлена на рисунке 2.

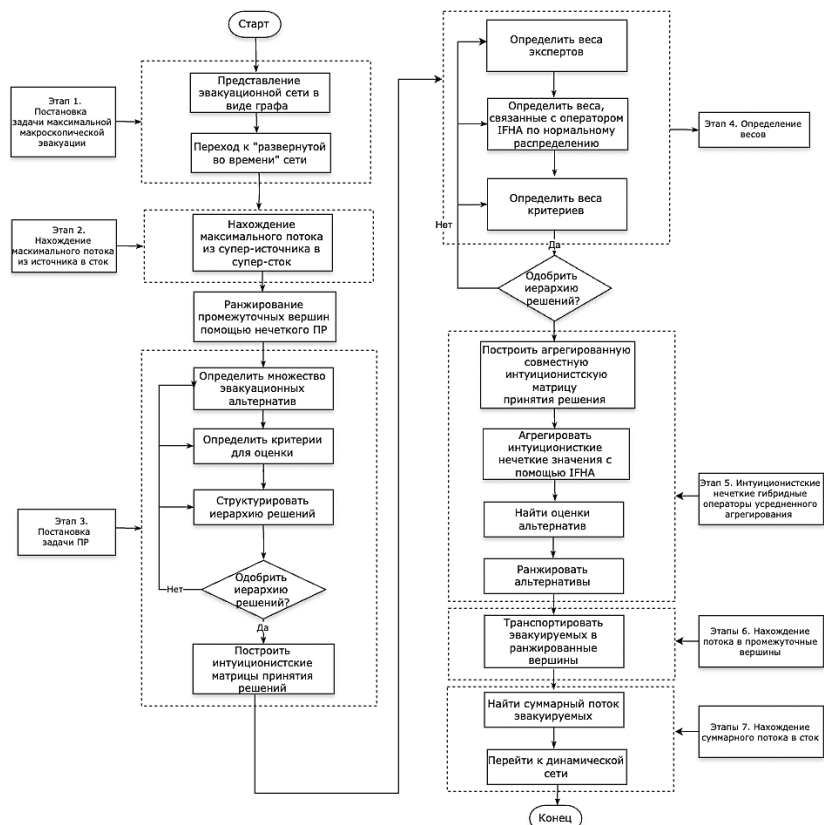


Рис. 2. Структурная схема интегрированного алгоритма принятия нечетких эвакуационных решений

Представленный алгоритм включает в себя следующие этапы. На первом этапе эвакуационная сеть представляется в виде динамического графа, и осуществляется переход к «развернутой во времени» сети. На втором этапе ищется максимальный поток из источника в стоки. Для транспортировки потока, оставшегося в промежуточных вершинах, реализуется перемещение людей из вершин-не стоков. Перемещение осуществляется из вершин, проранжированных по различным критериям согласно нечеткому

групповому принятию решений (этап 3). Для реализации этих целей вычисляются веса экспертов и критериев, а также веса, связанные с используемым оператором агрегирования IFНА (этап 4). На пятом этапе строятся агрегированные матрицы и применяются модифицированные операторы IFНА для оценки альтернатив. После нахождения списка промежуточных вершин на шестом этапе осуществляется транспортировка потока в сток из промежуточных вершин. Итоговый эвакуационный поток вычисляется на седьмом этапе.

Рассмотрим работу предложенного интегрированного алгоритма, включающего в себя нечеткое групповое принятие решений на основе модифицированного оператора IFНА, позволяющего работать с нечеткими интуиционистскими весами критериев, и нечеткую динамическую макроскопическую эвакуацию для транспортировки максимального числа людей в безопасные зоны более подробно.

Ввод исходных данных: Дана нечеткая динамическая сеть $\tilde{G} = (X, \tilde{A}, \tilde{u}_{ij}, \tilde{u}(x_i), \tau, S, I, D, T)$, $D = \{d_1, d_2, \dots, d_r\}$, $I = \{x_1, x, \dots, x_t\}$, $T = \{0, 1..T\}$, где $\tilde{u}_{ij} = \tilde{u}(a)$ с одним источником и несколькими стоками.

1. Производится переход к «расширенной во времени» статической сети $\tilde{G}^e = (X, \tilde{E}, \tilde{u}^e(\theta), \tau^e(\theta), S, I, D, T)$, $D = \{d_1, d_2, \dots, d_r\}$, $I = \{x_1, x, \dots, x_t\}$, $T = \{0, 1..T\}$, где $X^e = x_i(\theta) | x_i \in X; \theta = 0, 1, 2..T$. «Развернутое во времени» множество дуг $\tilde{E}^e$ определяется как $\tilde{E}^e = x_i(\theta), x_j(\vartheta) | (x_i, x_j) \in \tilde{E}; \theta = \vartheta + \tau_{ij}(\theta) \leq T; \theta = 0, 1, 2..T$. Добавляется супер-источник и супер-сток для соединения с несколькими источниками и одним стоком в каждый период времени. Введенные дуги имеют бесконечные дуговые способности.

2. Находится оптимальный порядок транспортировки пострадавших в безопасные места и промежуточные узлы [42 – 44]:

2.1. Определяется оптимальный порядок транспортировки пострадавших в безопасные места во время эвакуации. Здесь источники, стоки или промежуточные узлы для эвакуации задаются конечным набором альтернатив $A = \{A_1, A_2, \dots, A_n\}$, а экспертам $D_k, (k = 1, 2, \dots, s)$, принимающим решения, назначаются веса важности ω_k , где $\omega_k \geq 0, m(k = 1, 2, \dots, s), \sum_{k=1}^s \omega_k = 1$. Эксперты D_k оценивают альтернативы согласно критериям, представляя оценки в виде IFNs.

2.2. Осуществляется нормализация критериев. А также значения типа «затрат» преобразуются в значения типа «выгод» и

строится новая интуиционистская нечеткая матрица решений $R' = (r'_{ij})_{n \times m}$:

$$r_{ij} = (\mu_{ij}, \nu_{ij}) = \begin{cases} r'_{ij}, \text{ для атрибутов типа "выгоды",} \\ -\bar{r}'_{ij}, \text{ для атрибутов типа "затраты",} \end{cases} \quad (33)$$

где $\bar{r}'_{ij}$ – дополнение r'_{ij} , такое что $\bar{r}'_{ij} = (r_n, r_m)$.

2.3. Вычисляются веса экспертов на основе лингвистических термов. Пусть $\tilde{A}_e = \{\mu_{\tilde{A}_e}, \nu_{\tilde{A}_e}, \pi_{\tilde{A}_e}\}$ – интуиционистская нечеткая оценка e -ого эксперта. тогда вес эксперта ω_e определяется по формуле:

$$\omega_e = \frac{[\mu_e + \pi_e(\mu_e/(\mu_e + \nu_e))]}{\sum_{e=1}^k [\mu_e + \pi_e(\mu_e/(\mu_e + \nu_e))]} \quad (34)$$

где $\omega_e \in [0,1]$ и μ_e – оценка степени принадлежности e -ого эксперта, ν_e – оценка степени непринадлежности e -ого эксперта, π_e – оценка урвнения сомнения e -ого эксперта.

2.4. Реализуется механизм, основанный на нормальном распределении f [26, 36] для вычисления взвешенного вектора, используемого в операторе IFNA, $w = (w_1, w_2, \dots, w_n)^T$, $w_j \in [0,1]$, $j=(1,2,\dots,n)$ и $\sum_{j=1}^n w_j = 1$.

Из литературных источников и в мировой практике ученые из различных стран [26, 45 – 46] используют метод на основе нормального распределения для вычисления весов. Данный метод является широко используемым для вычисления весов в принятии решений, так как его реализация позволяет устранить предвзятость или предпочтения экспертов, влияющие на конечные результаты за счет высоких или низких значений этих аргументов, что приводит к некорректным оценкам.

Заметим, что многокритериальное групповое принятие решений согласно [36, 47 – 48] требует опыта и знаний экспертов. Для этого формируется набор из n значений предпочтений. При этом предвзятость или предпочтения экспертов влияют на конечные результаты за счет высоких или низких значений этих аргументов, что приводит к некорректным оценкам. Чтобы предотвратить эту проблему, эксперты назначают низкие веса субъективным аргументам. Таким образом, чем ближе значение предпочтения к медианным значениям, тем больше вес, и, наоборот, чем дальше значение

предпочтения от медианного значения, тем меньше вес. Данный механизм реализуется следующим образом.

Пусть $w = (w_1, w_2, \dots, w_n)^T$ – вектор весов оператора OWA, тогда справедливо следующее равенство:

$$w_i = \frac{1}{\sqrt{2\pi\sigma_n}} e^{-[(i-\mu_n)^2/2\sigma_n^2]}, i = 1, 2, \dots, n, \quad (35)$$

где μ_n – среднее значение совокупности из n чисел и $\sigma_n > 0$ – стандартное отклонение совокупности. μ_n и σ_n находятся по формулам:

$$\mu_n = \frac{1}{n} \frac{n(1+n)}{2} = \frac{1+n}{2}. \quad (36)$$

$$\sigma_n = \sqrt{\frac{1}{n} \sum_{i=1}^n (i - \mu_n)^2} \quad (37)$$

Учитывая, что $w_j \in [0, 1]$ и $\sum_{j=1}^n w_j = 1$, справедливо равенство:

$$w_i = \frac{\frac{1}{\sqrt{2\pi\sigma_n}} e^{-[(i-\mu_n)^2/2\sigma_n^2]}}{\sum_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_n}} e^{-[(j-\mu_n)^2/2\sigma_n^2]}} = \frac{e^{-[(i-\mu_n)^2/2\sigma_n^2]}}{\sum_{j=1}^n e^{-[(j-\mu_n)^2/2\sigma_n^2]}}, i = 1, 2, \dots, n. \quad (38)$$

2.5. Определяются веса критериев на основе нечеткого интуиционистского оператора взвешенного усреднения (IFWA). Если $w_j^k = \{\mu_j^k, \nu_j^k, \pi_j^k\}$ – интуиционистская нечеткая оценка j -го критерия k -м лицом, принимающим решение, то W представляет собой общий уровень важности всех экспертов. Вес критерия в виде лингвистических термов определяется на основе следующего выражения:

$$\begin{aligned} w_j &= IFWA(w_j^1, w_j^2, \dots, w_j^e) = \bigoplus_{e=1}^k \omega_e w_j \\ &= \left(1 - \prod_{e=1}^k (1 - \mu_{ij}^e)^{\omega_e}, \prod_{e=1}^k (\nu_{ij}^e)^{\omega_e} \right). \end{aligned} \quad (39)$$

2.6. Строится агрегированная коллективная нечеткая интуиционистская матрица решений (AIFDM) из индивидуальных исходных интуиционистских нечетких матриц решений (IFDM), значения которых представлены в виде лингвистических термов. Агрегированная коллективная матрица состоит из общей оценки альтернатив. Пусть $R^{(e)} = (r_{ij}^{(e)})_{m \times n}$ – IFDM каждого эксперта и $\omega = \{\omega_1, \omega_2, \omega_3, \dots, \omega_k\}$ – это вес эксперта, тогда:

$$r_{ij} = IFWA(r_{ij}^1, r_{ij}^2, \dots, r_{ij}^e) = \bigoplus_{e=1}^k \omega_e r_{ij} = \left(1 - \prod_{e=1}^k (1 - \mu_{ij}^e)^{\omega_e}, \prod_{e=1}^k (\nu_{ij}^e)^{\omega_e} \right). \quad (40)$$

2.7. Реализуется модифицированный оператор $IFHA$, учитывающий интуиционистские нечеткие веса критериев для нахождения нечетких оценок альтернатив:

$$r_{ij} = IFHA_{\tau, w}(\alpha_1, \alpha_2, \dots, \alpha_n) = \left(1 - \prod_{j=1}^n (1 - \mu_{\hat{\alpha}_{\sigma(j)}})^{w_j} \prod_{j=1}^n \nu_{\hat{\alpha}_{\sigma(j)}}^{w_j} \right), \text{ where } \hat{\alpha}_j = n\tau_j\alpha_j = \left(1 - \prod_{j=1}^n (1 - \mu_{\alpha_j})^{n\lambda_j}, \prod_{j=1}^n (\nu_{\alpha_j})^{n(1-\theta_j)} \right), \tau_j = (\lambda_j, \theta_j). \quad (41)$$

2.8. Вычисляются оценки $s(r_i)$ агрегированных интуиционистских значений согласно уравнениям (3)-(4).

2.9. Производится ранжирование этих оценок $s(r_i)$ для нахождения оптимального порядка убежищ для эвакуации. Если оценки $s(r_i)$ равны $s(r_j)$, то реализуется уравнение (5).

3. Производится приоритетное упорядочивание: $\forall x_k \in I$, определение их порядка и вычисление ранга на основе шагов 2.1–2.9.

Выход первого этапа алгоритма: ранжированные вершины графа.

4. Осуществляется переход к модифицированной динамической сети $\tilde{G}_m = (X_m, \tilde{A}_m, \tilde{u}_{ij}, \tilde{u}_m(x_i), \tau_{ij}, S_m, d_{1,m}, D_s, T)$, $\forall x_i \in I, \tilde{u}(x_i) > 0$ с тем же приоритетом, что соответствующие им промежуточные вершины, $\tilde{u}_m(x_i) = \tilde{u}(x_i)$, $\tau(x_i) = 0$. Добавление супер-стока D^* , связывающего все искусственные стоки с ним.

5. Осуществляется переход к развернутой во времени статической сети $\tilde{G}_m^* = (X_m^*, \tilde{A}_m^*, \tilde{u}_{ij}^*, \tau_{ij}, S_m^*, I^*, d_{1,m}^*, D_s^*, T)$ на основе создания расширенной во времени копии каждой вершины.

6. Вычисляется максимальный поток в развернутой во времени сети $\tilde{G}_m^*$ без промежуточного хранения:

для $\forall (x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*})) \in \tilde{A}_m^*$ **выполнить:**

$$\tilde{v}(x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*})) \leftarrow 0.$$

$$\tilde{v}(x_2^*(\theta + \tau_{x_1^* x_2^*}), x_1^*(\theta)) \leftarrow 0.$$

Пока существует путь от s до супер-стока без циклов в остаточной сети $\tilde{G}_{mr}^*$, **выполнить:**

$$\delta \leftarrow \min \{ \tilde{u}_r^* (x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*})) : (x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*})) \in p \}.$$

Для каждого ребра $(x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*}))$ в p выполнить

$$\tilde{v}(x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*})) \leftarrow \tilde{v}(x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*})) + \delta$$

$$\tilde{v}(x_2^*(\theta + \tau_{x_1^* x_2^*}), x_1^*(\theta)) \leftarrow -\tilde{v}(x_1^*(\theta), x_2^*(\theta + \tau_{x_1^* x_2^*})).$$

7. Производится переход к сети $\tilde{G}_m$ от $\tilde{G}_m^*$ путем удаления супер-стока и фиктивных вершин.

8. Находится максимальный поток от источника до промежуточных узлов в соответствии с порядком приоритетов, найденным на шаге 3 и шагах 2.1–2.9.

9. Производится декомпозиция потока на пути.

10. Осуществляется переход к динамической сети.

11. *Выход:* динамическая сеть с максимальным потоком.

Таким образом, разработанный алгоритм оперирует тремя видами весов: весами атрибутов (критериев), весами экспертов и весами, связанными с оператором IFNA.

Заметим, что в итоге реализации предложенного алгоритма получаем динамическую сеть с максимальным эвакуационным потоком.

6. Моделирование и обсуждение результатов принятия решений при эвакуации. Для подтверждения эффективности разработанного алгоритма создана программная среда, реализованная на языке JavaScript. На рисунке 3 приведена архитектура программного модуля и входных/выходных данных. Входные данные в виде файлов передаются на вход программного модуля через аргументы командной строки. Выходные файлы сохраняются в директории исполняемого файла программного модуля.

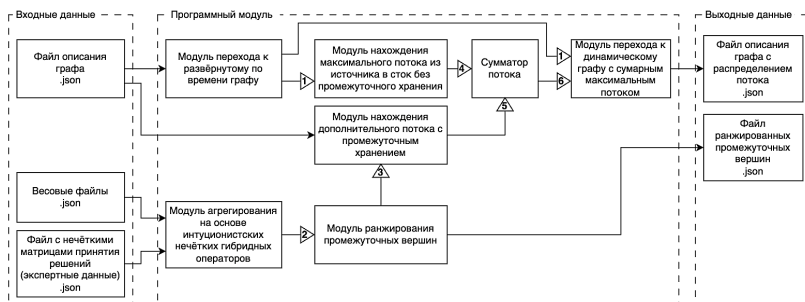


Рис. 3. Архитектура программного модуля

Потоки данных между модулями на рисунке 3 обозначены цифрами. В частности, 1 – «развернутый во времени» граф; 2 – агрегированная коллективная интуиционистская нечеткая матрица принятия решений и оценки альтернатив в виде IFNs; 3 – множество ранжированных вершин-убежищ; 4 – величина максимального потока без промежуточного хранения; 5 – дополнительный поток; 6 – суммарный поток.

Проведено моделирование эвакуации из здания для максимизации числа спасенных жизней. На рисунке 4 представлен пример здания для эвакуации.

Для выполнения принятия решений необходимо использовать не сами здания, а их структурные модели. Данные модели легко интерпретируются в виде графов и в их матричных и списковых эквивалентах. На рисунке 5 построена графовая модель здания с опасной зоной-источником (s) и тремя убежищами (x_6, x_7, x_8).

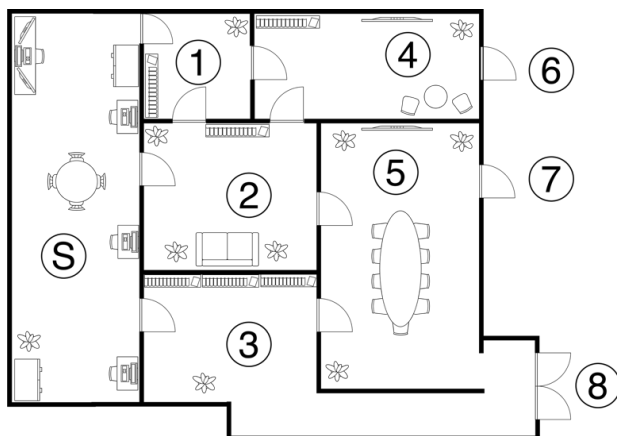


Рис. 4. Пример здания

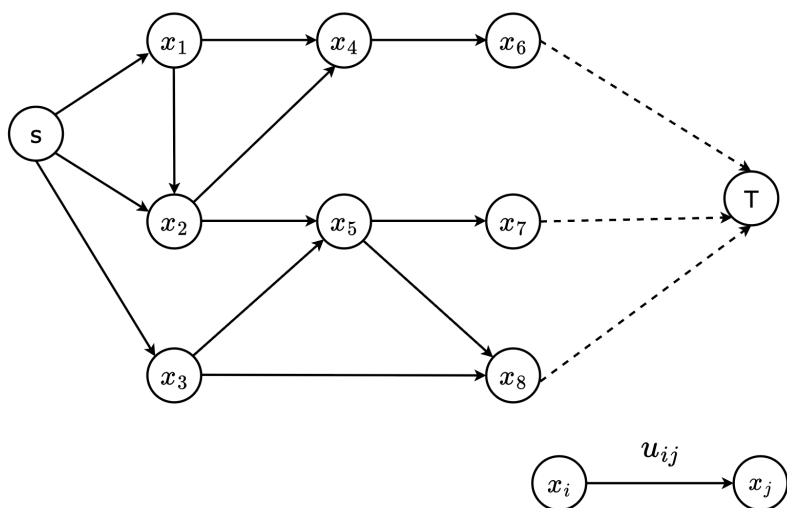


Рис. 5. Графовая модель здания

Заметим, что пропускные способности дуг сети – коридоров – могут изменяться во времени из-за перекрытия коридоров огнём или дымом. Помещения x_1 , x_2 , x_3 , x_4 имеют ограниченные пропускные способности, которые также могут изменяться во времени. В этих помещениях размещают людей с целью их дальнейшего перемещения в безопасные места, что позволяет увеличить число спасённых жизней. Время прохождения эвакуационного потока по дугам сети является

динамической величиной. Пропускные способности дуги сети задаются в нечётком виде (в виде центра нечеткого числа), так как они оцениваются лишь приблизительно в связи с возникновением ЧС. Далее находятся левые и правые коэффициенты размаха (коэффициенты нечеткости, индексы нечеткости) по линейным комбинациям соседних значений, что не приводит к сильному размытию границ числа. Таблицы 1-3 иллюстрируют нечеткие пропускные способности дуг, вершин сети и параметры времени прохождения потока.

Таблица 1. Нечеткие пропускные способности дуг сети

	Динамические пропускные способности дуг и промежуточных вершин				
	$\theta = 0$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
(x_i, x_j)	u_{ij}				
(s, x_1)	$\overline{20}$	$\overline{23}$	$\overline{25}$	$\overline{20}$	$\overline{23}$
(s, x_2)	$\overline{105}$	$\overline{120}$	$\overline{100}$	$\overline{90}$	$\overline{100}$
(s, x_3)	$\overline{90}$	$\overline{86}$	$\overline{86}$	$\overline{90}$	$\overline{92}$
(x_1, x_2)	$\overline{15}$	$\overline{15}$	$\overline{13}$	$\overline{15}$	$\overline{15}$
(x_1, x_4)	$\overline{10}$	$\overline{10}$	$\overline{12}$	$\overline{12}$	$\overline{12}$
(x_2, x_4)	$\overline{80}$	$\overline{85}$	$\overline{80}$	$\overline{80}$	$\overline{90}$
(x_2, x_5)	$\overline{60}$	$\overline{65}$	$\overline{65}$	$\overline{58}$	$\overline{58}$
(x_3, x_5)	$\overline{42}$	$\overline{42}$	$\overline{48}$	$\overline{48}$	$\overline{45}$
(x_3, x_8)	$\overline{66}$	$\overline{70}$	$\overline{70}$	$\overline{66}$	$\overline{66}$
(x_4, x_6)	$\overline{40}$	$\overline{30}$	$\overline{35}$	$\overline{40}$	$\overline{35}$
(x_5, x_7)	$\overline{55}$	$\overline{50}$	$\overline{45}$	$\overline{55}$	$\overline{55}$
(x_5, x_8)	$\overline{44}$	$\overline{40}$	$\overline{44}$	$\overline{40}$	$\overline{42}$

Таблица 2 отражает ненулевые пропускные способности промежуточных вершин.

Таблица 2. Нечеткие пропускные способности промежуточных вершин сети

	Динамические пропускные способности дуг и промежуточных вершин				
	$\theta = 0$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
x_i	u_i				
x_1	$\overline{10}$	$\overline{15}$	$\overline{15}$	$\overline{10}$	$\overline{10}$
x_2	$\overline{20}$	$\overline{20}$	$\overline{23}$	$\overline{23}$	$\overline{20}$
x_3	$\overline{25}$	$\overline{25}$	$\overline{23}$	$\overline{23}$	$\overline{25}$
x_4	$\overline{18}$	$\overline{18}$	$\overline{18}$	$\overline{16}$	$\overline{16}$

Таблица 3. Параметры времени прохождения потока по дугам для сети

	Динамические пропускные параметры времени				
	$\theta = 0$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
(x_i, x_j)	τ_{ij}				
(s, x_1)	1	1	2	2	2
(s, x_2)	1	1	1	2	2
(s, x_3)	2	2	1	1	1
(x_1, x_2)	2	1	1	1	1
(x_1, x_4)	3	1	1	1	2
(x_2, x_4)	2	1	1	1	2
(x_2, x_5)	2	2	2	1	2
(x_3, x_5)	2	1	1	1	3
(x_3, x_8)	2	3	2	2	2
(x_4, x_6)	1	1	1	1	2
(x_5, x_7)	2	2	1	1	1
(x_5, x_8)	2	1	1	1	2

После построения сети осуществляется поиск максимального эвакуационного потока из источника в стоки без промежуточного размещения пострадавших в вершинах-не стоках согласно шагам Алгоритма 4-7, что отражено в таблице 4.

Таблица 4. Общий поток без промежуточного хранения

Динамический путь	Статические пути	Статический поток	Время отправления	Время прибытия	Динамический поток	Сток
1) $s \rightarrow x_2 \rightarrow x_4 \rightarrow x_6$	1) $s_0 \rightarrow x_{2,1} \rightarrow x_{4,2} \rightarrow x_{6,3}$	$\overline{35}$	0	3	$\overline{75}$	x_6
	2) $s_1 \rightarrow x_{2,2} \rightarrow x_{4,3} \rightarrow x_{6,4}$	$\overline{40}$	1	4		
2) $s \rightarrow x_2 \rightarrow x_5 \rightarrow x_8$	1) $s_0 \rightarrow x_{2,1} \rightarrow x_{5,3} \rightarrow x_{8,4}$	$\overline{40}$	0	4	$\overline{40}$	x_8
3) $s \rightarrow x_2 \rightarrow x_5 \rightarrow x_7$	1) $s_0 \rightarrow x_{2,1} \rightarrow x_{5,3} \rightarrow x_{7,4}$	$\overline{25}$	0	4	$\overline{25}$	x_7
4) $s \rightarrow x_3 \rightarrow x_5 \rightarrow x_7$	1) $s_0 \rightarrow x_{3,2} \rightarrow x_{5,3} \rightarrow x_{7,4}$	$\overline{30}$	0	4	$\overline{30}$	x_7
5) $s \rightarrow x_3 \rightarrow x_8$	1) $s_0 \rightarrow x_{3,2} \rightarrow x_{8,4}$	$\overline{60}$	0	4	$\overline{60}$	x_8
Общий поток без промежуточного хранения					$\overline{230}$	

Результаты работы этих шагов Алгоритма представлены на рисунке 6.

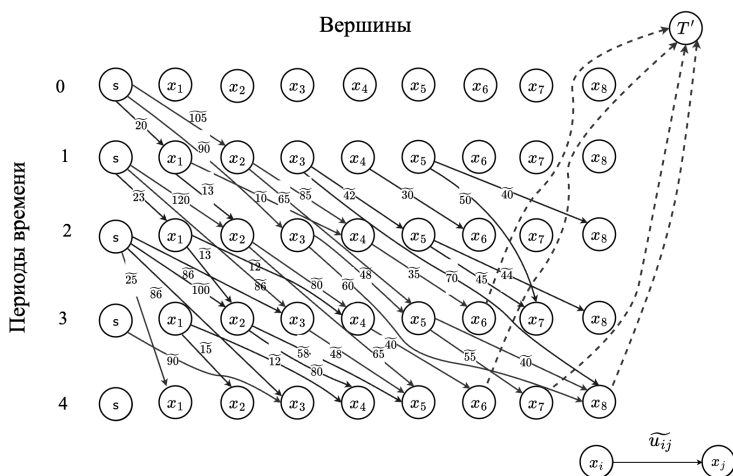


Рис. 6. Динамическая сеть с общим потоком в стоки без промежуточного хранения

Производится ранжирование промежуточных вершин для передачи дополнительного потока эвакуируемых из этих вершин в стоки согласно шагам 2.1-2.9 Алгоритма.

Заметим, что четыре из пяти промежуточных вершин x_1, x_2, x_3, x_4 имеют ненулевые пропускные способности. Таким образом, s – вершина-источник; x_6, x_7, x_8 – стоки; x_1, x_2, x_3, x_4, x_5 – промежуточные вершины, из которых x_1, x_2, x_3, x_4 имеют возможность хранить поток и перевозить его дальше в сток. Группа экспертов представляет свои оценки для определения порядка альтернатив эффективной эвакуации. При этом рассматриваются следующие критерии: вместимость помещения, дающая верхний предел количества пострадавших, которые могут находиться в этом укрытии (x_1), легкость эвакуации (x_2), уровень безопасности помещения как убежища (x_3), время организации эвакуации из этого места (x_4) и отдаленность от источнику опасности (x_5). Все критерии относятся к типу выгоды.

В таблицах 5–7 показаны лингвистические термы (наборы лингвистических переменных) для оценки экспертов, атрибутов и

альтернатив в виде интуиционистских нечетких чисел. Таблица 8 иллюстрирует веса важности критериев в виде лингвистических термов.

Таблица 5. Лингвистические термы для оценки экспертов

Лингвистический терм	Интуиционистское нечеткое число
Очень низко (ОН)	(0.10, 0.80)
Низко (Н)	(0.20, 0.70)
Средне-низко (СН)	(0.35, 0.50)
Средне (С)	(0.40, 0.55)
Средне-важно (СВ)	(0.6, 0.35)
Важно (В)	(0.70, 0.20)
Очень важно (ОВ)	(0.90, 0.05)

Таблица 6. Лингвистические термы для оценки критериев

Лингвистический терм	Интуиционистское нечеткое число
Очень слабо (ОС)	(0.15, 0.80)
Слабо (С)	(0.30, 0.65)
Ниже среднего (НС)	(0.45, 0.50)
Средне (Ср)	(0.55, 0.40)
Средне-важно (СВ)	(0.65, 0.30)
Важно (В)	(0.75, 0.15)
Очень важно (ОВ)	(0.85, 0.10)

Таблица 7. Лингвистические термы для оценки альтернатив

Лингвистический термин	Интуиционистское нечеткое число
Очень низко (ОН)	(0.10, 0.85)
Низко (Н)	(0.15, 0.75)
Средне-низко (СН)	(0.25, 0.60)
Средне (С)	(0.50, 0.50)
Средне-хорошо (СХ)	(0.7, 0.25)
Хорошо (Х)	(0.80, 0.10)
Очень хорошо (ОХ)	(0.9, 0.05)

Таблица 8. Веса важности критериев

	x_1	x_2	x_3	x_4	x_5
D_1	ОС	С	НС	В	В
D_2	С	СВ	Ср	СВ	ОВ
D_3	Ср	В	СВ	С	Ср
D_4	НС	ОВ	В	Ср	НС

Веса лиц, принимающих решения, основаны на лингвистических терминах в таблице 5, и вычисляются с помощью уравнения (34). В результате получены следующие веса важности (таблица 9):

Таблица 9. Важность и вес оценок лиц, принимающих решения (ЛПР)

	ЛПР1	ЛПР2	ЛПР3	ЛПР4
Лингвистический термин	Важный	Средне-важный	Очень важный	Средний
ω	0.28	0.227	0.34	0.152

Затем вычисляется вектор весов методом нормального распределения, чтобы исключить влияние чрезмерно низких и высоких весов по уравнениям (35)–(38).

$$\mu_n = 3, \sigma_n = \sqrt{2}.$$

$$w = (0.1117, 0.2365, 0.3036, 0.2365, 0.1117)^T.$$

На основе данных из таблицы 9 и уравнения (39) определены веса важности критериев в виде IFNs (таблица 10).

Таблица 10. Вес критерия

x_1	(0.387, 0.561, 0.052)
x_2	(0.667, 0.249, 0.084)
x_3	(0.600, 0.333, 0.067)
x_4	(0.581, 0.336, 0.083)
x_5	(0.693, 0.229, 0.077)

Заметим, что эвакуационные альтернативы оценивались лицами, принимающими решения, с учетом лингвистических термов из таблицы 7 по каждому критерию. Далее формируются нечёткие интуиционистские матрицы четырех экспертов (таблицы 11–14).

Таблица 11. Интуиционистская нечеткая матрица принятия решений первого эксперта

	x_1	x_2	x_3	x_4	x_5
A_1	C	CX	OX	H	CH
A_2	X	CH	OX	H	CX
A_3	CX	OX	C	CH	OH
A_4	X	CX	CH	H	OH

Таблица 12. Интуиционистская нечеткая матрица принятия решений второго эксперта

Альтернатива	Критерий				
	x_1	x_2	x_3	x_4	x_5
A_1	CH	CH	X	CH	H
A_2	OX	H	X	CH	C
A_3	C	X	CX	OH	CH
A_4	C	X	CX	CH	CH

Таблица 13. Интуиционистская нечеткая матрица принятия решений третьего эксперта

Альтернатива	Критерий				
	x_1	x_2	x_3	x_4	x_5
A_1	CH	H	OX	H	CH
A_2	X	CH	OX	OH	X
A_3	CX	OX	C	H	CX
A_4	CX	OX	CH	CX	C

Таблица 14. Интуиционистская нечеткая матрица принятий решений четвертого эксперта

Альтернатива	Критерий				
	x_1	x_2	x_3	x_4	x_5
A_1	C	C	CH	OX	H
A_2	CH	H	X	H	CX
A_3	X	C	CX	H	X
A_4	X	CH	C	OX	CH

Далее на основе данных из таблиц 11-14 и уравнения (40) строится агрегированная коллективная интуиционистская нечеткая матрица принятия решений (AIFDM), которая приведена в таблице 15.

С учетом модифицированного оператора *IFHA* (уравнение 41) и данных из таблицы 15 получены оценки альтернатив в виде IFNs, что показано в таблице 16.

Таблица 15. Агрегированная коллективная интуиционистская нечеткая матрица принятия решений (AIFDM)

	x_1	x_2	x_3	x_4	x_5
A_1	(0.370, 0.555, 0.075)	(0.431, 0.493, 0.077)	(0.841, 0.085, 0.074)	(0.403, 0.473, 0.124)	(0.214, 0.653, 0.133)
A_2	(0.791, 0.112, 0.097)	(0.214, 0.653, 0.133)	(0.870, 0.065, 0.065)	(0.158, 0.744, 0.098)	(0.707, 0.214, 0.079)
A_3	(0.683, 0.255, 0.062)	(0.851, 0.083, 0.066)	(0.588, 0.384, 0.028)	(0.169, 0.725, 0.107)	(0.527, 0.374, 0.099)
A_4	(0.717, 0.197, 0.086)	(0.784, 0.134, 0.082)	(0.427, 0.478, 0.094)	(0.581, 0.325, 0.094)	(0.313, 0.622, 0.0655)

Таблица 16. IFNA

A_1	(0.687, 0.180, 0.133)
A_2	(0.890, 0.035, 0.855)
A_3	(0.871, 0.067, 0.804)
A_4	(0.863, 0.057, 0.806)

Затем осуществляется вычисление оценок $s(r_i)$ агрегированных интуиционистских значений с учетом уравнения (4):

$$s(r_1) = 0.507, s(r_2) = 0.855, s(r_3) = 0.804, s(r_4) = 0.806.$$

Далее производится ранжированы альтернатив:

$$s(r_2) > s(r_4) > s(r_3) > s(r_1).$$

Это значит, что $x_2 > x_4 > x_3 > x_1$.

Осуществлена передача дополнительного потока из промежуточных вершин в порядке $x_2 > x_4 > x_3 > x_1$ в стоки для нахождения максимального потока эвакуируемых. Тогда максимальный динамический поток с промежуточным хранением представлен в таблице 17.

Таблица 17. Максимальный динамический поток с промежуточным хранением

Динамический путь	Статические пути	Статический поток	Время отправления	Время прибытия	Динамический поток	Сток
Общий поток без промежуточного хранения					$\overline{230}$	
1) $s \rightarrow x_2 \rightarrow x'_2$	1) $s_0 \rightarrow x_{2,1} \rightarrow x'_{2,1}$	$\overline{5}$	0	1		
	2) $s_1 \rightarrow x_{2,2} \rightarrow x'_{2,2}$	$\overline{23}$	1	2	$\overline{51}$	x_2
	3) $s_2 \rightarrow x_{2,3} \rightarrow x'_{2,3}$	$\overline{23}$	2	3		
2) $s \rightarrow x_1 \rightarrow x_4$	1) $s_0 \rightarrow x_{1,1} \rightarrow x_{4,2} \rightarrow x'_{4,1}$	$\overline{10}$	0	1		
	2) $s_1 \rightarrow x_{2,2} \rightarrow x_{4,3} \rightarrow x'_{4,3}$	$\overline{16}$	1	3	$\overline{42}$	x_4
	3) $s_2 \rightarrow x_{2,3} \rightarrow x_{4,4} \rightarrow x'_{4,4}$	$\overline{16}$	2	4		
	1) $s_1 \rightarrow x_{3,3} \rightarrow x'_{3,3}$	$\overline{23}$	1	3		
3) $s \rightarrow x_3$	2) $s_2 \rightarrow x_{3,4} \rightarrow x'_{3,4}$	$\overline{25}$	2	4	$\overline{48}$	x_3
	1) $s_1 \rightarrow x_{1,2} \rightarrow x'_{1,2}$	$\overline{15}$	1	2	$\overline{25}$	x_1
4) $s \rightarrow x_1$	2) $s_2 \rightarrow x_{1,4} \rightarrow x'_{1,4}$	$\overline{10}$	2	4		
Поток с промежуточным хранением					$\overline{166}$	
Общий максимальный поток					$\overline{396}$	

Таким образом, исходя из таблицы 17, максимальный поток эвакуации без промежуточного размещения пострадавших составляет $\overline{230}$ единиц, в то время как с возможностью промежуточного размещения – $\overline{396}$ единиц. На рисунке 7 представлено это распределение потока эвакуации.

Далее находятся коэффициенты размаха нечеткого числа $\overline{396}$ согласно методу, описанному в работах [41 – 42], который основан на линейных комбинациях индексов нечеткости. Для расчета индексов нечеткости искомого числа необходимое значение определяется по соседним значениям индексов (рисунок 8).

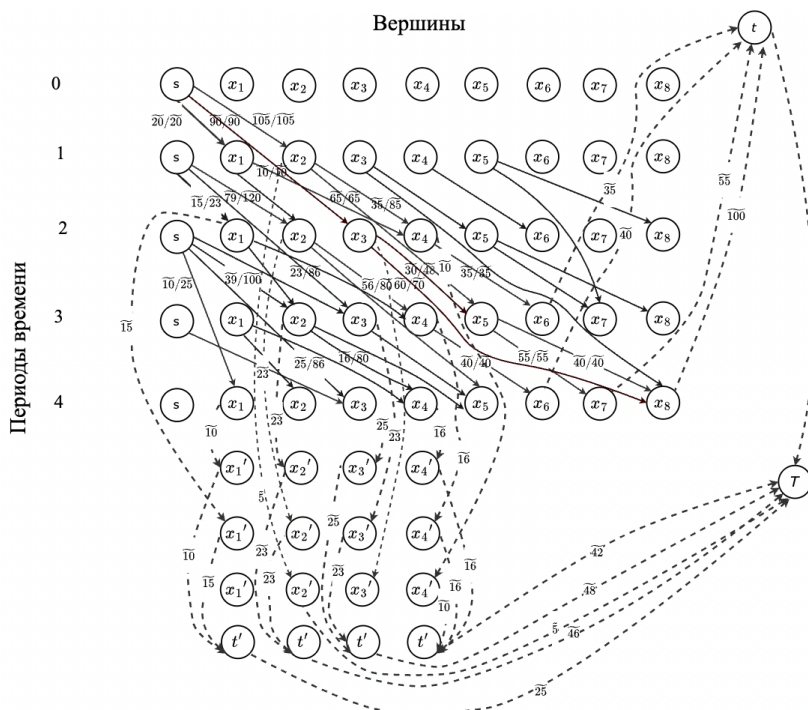
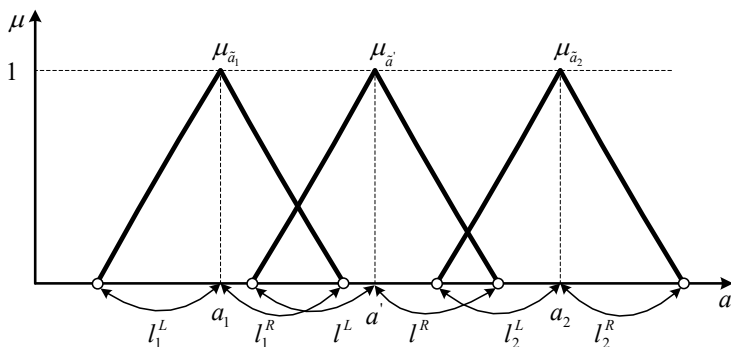


Рис. 7. Максимальный поток с промежуточным хранением

Рис. 8. Определение функции принадлежности $\mu_{a'}(a)$

Пусть нечеткий параметр «около a'' » находится между двумя соседними значениями «около a_1 » и «около a_2 » ($a_1 \leq a' \leq$

a_2), функции принадлежности которого $\mu_{a_1}(a_1)$ и $\mu_{a_2}(a_2)$ имеют треугольную форму. Следовательно, коэффициенты размаха функции принадлежности $\mu_{a'}(a)$, соответствующие нечеткому параметру «около $\tilde{x}'$ », задаются линейной комбинацией левого и правого индексов нечеткости соседних значений:

$$l^L = \frac{(a_2 - a')}{(a_2 - a_1)} \times l_1^L + \left(1 - \frac{(a_2 - a')}{(a_2 - a_1)}\right) \times l_2^L, \quad (42)$$

$$l^R = \frac{(a_2 - a')}{(a_2 - a_1)} \times l_1^R + \left(1 - \frac{(a_2 - a')}{(a_2 - a_1)}\right) \times l_2^R. \quad (43)$$

где l^L – левый коэффициент размаха нечеткого треугольного числа с центром a_1 ; l^R – правый коэффициент размаха нечеткого треугольного числа с центром a_2 . В частном случае, когда центр искомого треугольного числа равен значению, существующему на числовой оси, размахи нечеткого числа совпадают с размахами существующего числа. Если центр нечеткого параметра расположен левее центра первого значения на числовой оси, его коэффициенты размаха совпадают с теми, которые расположены на оси. В противном случае, когда центр искомого числа находится справа от центра последнего числа, отмеченного на оси, их коэффициенты размаха совпадают.

Учитывая вышеизложенное, были найдены коэффициенты размаха нечеткого числа $\widetilde{396}$ по формулам (42-43) на основе базовых значений нечетких чисел, заданных экспертами (рисунок 9).

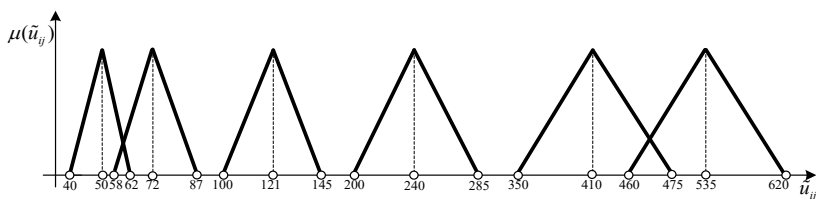


Рис. 9. Базовые значения функций принадлежности пропускных способностей дуг сети

Требуемое значение потока находится между двумя соседними базовыми значениями: $\widetilde{240}$ с левым коэффициентом нечеткости $l_1^L = 40$, правым коэффициентом нечеткости – $l_1^R = 45$ и $\widetilde{410}$ с левым

коэффициентом нечеткости $l_1^L = 60$, правым коэффициентом нечеткости – $l_1^R = 65$. На основании уравнения (7): $l_1^L = 54.68$, $l_1^R = 59.69$. Наконец, максимальный поток эвакуации в динамической нечеткой сети с частичным разворотом полосы движения может быть представлен единицами нечеткого треугольного числа (311, 365, 424).

Так как пропускные способности дуг оценены лишь приблизительно, в нечетких числах, то и значение потока будет выражаться нечетким треугольным числом. Мы можем интерпретировать его следующим образом: максимальное число эвакуируемых со степенью уверенности 1 оценивается в диапазоне от 311 до 424, со степенью уверенности 0,8 оно измеряется в диапазоне от 354 до 377, вероятнее всего оно будет равно 365, но в любом случае значение максимального потока не может быть меньше 311 и больше 424 (рисунок 10).

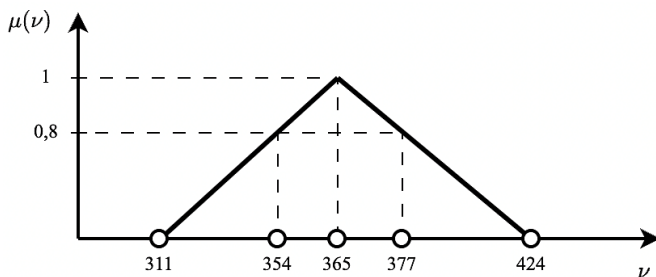


Рис. 10. Нечеткое число и альфа-срез

Относительно неточности Лотфи Заде говорил: «Я считаю, что излишнее стремление к точности стало оказывать действие, сводящее на нет теорию управления и теорию систем, так как оно приводит к тому, что исследования в этой области сосредотачиваются на тех и только тех проблемах, которые поддаются точному решению. В результате многие классы важных проблем, в которых данные, цели и ограничения являются слишком сложными или плохо определёнными для того, чтобы допустить точный математический анализ, оставались и остаются в стороне по той причине, что они не поддаются математической трактовке. Для того чтобы сказать что-либо существенное для проблем подобного рода, мы должны отказаться от наших требований точности и допустить результаты, которые являются несколько размытыми или неопределённым» [49].

Для проверки вычислительной сложности предложенного алгоритма был проведён вычислительный эксперимент (таблица 18).

Случайным образом были подготовлены тестовые графы на 10, 100, 1000, 5000 и 10000 вершин. Каждый граф содержал различное количество рёбер (от 80 до 200% рёбер в каждом). Графы генерируются случайным образом с обязательной проверкой на наличие путей от истока к стоку. Для каждого количества вершин сгенерировано по 1000 графов. Для каждого набора проведено по 1000 запусков. В таблице 5 приведена зависимость времени работы программного модуля от количества вершин. Приведены минимальное зарегистрированное время работы алгоритма, максимальное и усреднённое. При этом 10% самых низких и самых высоких значений в каждой группе были исключены из рассмотрения для устранения аномально быстрых и аномально долгих запусков. Все эксперименты проводились на ПК MacBook Air (M1, 2020), размер оперативной памяти 16 ГБ.

Таблица 18. Время работы программного модуля в зависимости от числа вершин.

Количество вершин, шт	Минимальное время, мс	Максимальное время, мс	Среднее время, мс
10	0,0122	0,029	0,0245
100	0,148	0,212	0,198
1000	0,821	1,045	0,921
5000	1,275	2,343	2,194
10000	3,448	10,595	8,430

Было проведено сравнение результатов предлагаемого алгоритма на основе модифицированного оператора IFHA с другими интуиционистскими операторами агрегирования, которые также были модифицированы с помощью уравнений (23)–(28) для интуиционистских нечетких чисел, что показано в таблице 19.

Таблица 19. IFHA и другие операторы

	$s(r_1)$	$s(r_2)$	$s(r_3)$	$s(r_4)$
IFHA	0.507	0.855	0.804	0.806
IFHG	−0.990	−0.989	−0.986	−0.983
IFWA	0.850	0.940	0.928	0.910
IFWG	−0.804	−0.769	−0.690	−0.657
IFOWA	−0.223	0.235	−0.108	0.137
IFOWG	0.054	0.270	0.364	0.355

Сравнение алгоритмов, основанных на различных операторах агрегирования, представлено на рисунке 11.

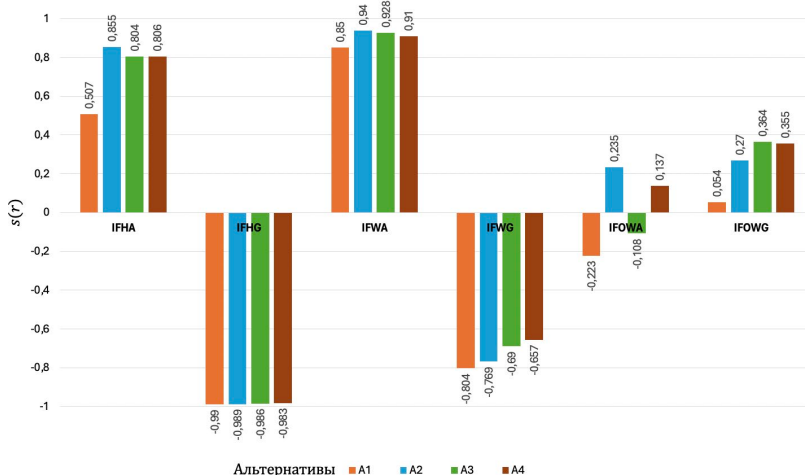


Рис. 11. Результаты алгоритмов IFNA и IFNG

На основе проведенного анализа построенных таблиц и графиков сделан вывод о достоверности предлагаемого метода принятия решения на основе оператора IFNA, оперирующего нечеткими весами критериев. Операторы IFWA и IFNA имеют одинаковый порядок альтернатив эвакуации: $A_2 > A_4 > A_3 > A_1$, что означает, что первой осуществляется эвакуация из вершины x_2 . Все рассмотренные операторы указали A_1 как худшую альтернативу. При этом все операторы усреднения выделили A_2 как лучшую и A_1 как худшую альтернативы с небольшим отличием между второй и третьей альтернативами, а геометрические операторы IFWG и IFNG показали одинаковые результаты и выделили A_4 как лучшую альтернативу и A_1 как худшую. При этом оператор IFOWG имеет отличие в первой и второй альтернативах, а IFWA выделил A_2 как лучшую альтернативу и A_1 как худшую.

К ограничениям алгоритма можно отнести его зависимость от точности экспертных оценок, связанную с сомнениями и неуверенностью экспертов при выборе функций принадлежности ввиду отсутствия статических данных или точной оценки параметров сети.

7. Заключение. В статье рассматривается задача эвакуации из здания на основе многокритериального принятия решений с учетом нечеткого интуиционистского гибридного оператора усреднения и стратегии макроскопической эвакуации. Для ее решения разработан интегрированный алгоритм эвакуации из здания, объединяющий нечеткое принятие решений на основе интуиционистского нечеткого гибридного оператора усреднения и нахождение лексикографического потока с учетом промежуточного размещения пострадавших в вершинах-не убежищах. Данный алгоритм является многоуровневым, где на первом уровне осуществляется построение модели задачи в виде эвакуационной транспортной сети. На втором уровне алгоритма производится ранжирование эвакуационных помещений для нахождения оптимального порядка транспортировки людей из них в убежища на основе различных критериев, таких как вместимость помещения, легкость эвакуации, уровень безопасности помещения как убежища, время организации эвакуации из этого места и удаленность от источника опасности. Ранжирование помещений осуществляется на основе многокритериального группового принятия решений и интуиционистского нечеткого гибридного оператора усреднения для моделирования сомнений и неуверенности. Алгоритм оперирует лингвистическими оценками экспертов, позволяя на их основе вычислять веса экспертов и критериев для эффективного принятия решения. Агрегирование оценок осуществляется на основе методики, которая позволяет оперировать весами критериев, представленными в виде нечетких интуиционистских значений в отличие от традиционных четких чисел для учета сомнений и неуверенности группы экспертов. На третьем уровне алгоритма осуществляется макроскопическая потоковая динамическая эвакуация для перемещения максимального количества потока из опасных зон с учетом возможности размещения пострадавших в промежуточных вершинах, не являющихся убежищами, в безопасные хранилища. Алгоритм позволяет учесть веса критериев и экспертов, представленные в нечетком лингвистическом виде. Для подтверждения эффективности разработанного алгоритма создана программная среда, реализованная на языке JavaScript. Проведено сравнение разработанного алгоритма принятия решения на основе оператора IFNA, оперирующего нечеткими весами критериев, с существующими алгоритмами и сделан вывод о достоверности предлагаемого алгоритма. Проведена оценка зависимости времени работы алгоритма от размера входных, подтверждающая возможность

использования предложенного алгоритма для больших зданий и транспортных сетей.

Результаты данного алгоритма могут применяться как при реальной эвакуации населения в случае масштабных ЧС, так и при составлении планов реагирования на эвакуацию. В перспективе планируется учитывать: панику и нерациональное поведение людей при эвакуации, наличие заблокированных путей (например, из-за обрушения зданий или деревьев на дорогах), параметры риска и опасности эвакуации, которые могут сделать «оптимальные» маршруты недоступными другие факторы.

Литература

1. Герасименко Е.М., Кравченко Д.Ю., Кравченко Ю.А., Кулиев Э.В. Поддержка принятия решений по предупреждению и ликвидации последствий чрезвычайных ситуаций на основе нечеткого метода структурирования информации // Известия ЮФУ. Технические науки. 2023. № 2(232). С. 201–212.
2. Shi D., Li J., Wang Q., Chen J., Lovreglio R., Lo J.T.Y., Ma J. Modeling and simulation of crowd dynamics at evacuation bottlenecks during flood disasters // Transportation Research Part E: Logistics and Transportation Review. 2025. vol. 198.
3. Feng Z., Wang C., An J., Zhang X., Liu X., Ji X., Kang L., Quan W. Emergency fire escape path planning model based on improved DDPG algorithm // Journal of Building Engineering. 2024. vol. 95. DOI: 10.1016/j.jobe.2024.110090.
4. Kim J., Kim B.-J., Kim N. Perception-based analytical technique of evacuation behavior under radiological emergency: An illustration of the Kori area // Nuclear Engineering and Technology. 2021. № 53(3). pp. 825–832. DOI: 10.1016/j.net.2020.08.012.
5. Ford J., Fulkerson D.R. Maximal flow through a network // Canadian Journal of Mathematics. 1956. № 8(3). pp. 399–404.
6. Gale D. Transient flows in networks // Michigan Mathematical Journal. 1959. № 6(1). pp. 59–63.
7. Minieka E. Maximal, lexicographic, and dynamic network flows // Oper. Res. 1973. № 21(2). pp. 517–527. DOI: 10.1287/opre.21.2.517.
8. Chalmet L.G., Francis R.L., Saunders P.B. Network models for building evacuation // Fire Technology. 1982. № 18(1). pp. 90–113. DOI: 10.1007/BF02993491.
9. Opananon S. On finding paths and flows in multicriteria, stochastic and time-varying networks // Ph.D. dissertation. University of Maryland. 2004.
10. Cui J., An S., Zhao M. A Generalized Minimum Cost Flow Model for Multiple Emergency Flow Routing. Mathematical Problems in Engineering. 2014. DOI: 10.1155/2014/832053.
11. Hamacher H.W., Tufekci S. On the use of lexicographic min cost flows in evacuation modeling // Naval Research Logistics. 1987. vol. 34(4). pp. 487–503.
12. Dunn C.E., Newton D. Optimal routes in GIS and emergency planning applications // Area. 1992. vol. 24(3). pp. 259–267.
13. Miller-Hooks E., Patterson S.S. On solving quickest time problems in time-dependent, dynamic networks // Journal of Mathematical Modelling and Algorithms. 2004. № 3(1). pp. 39–71.
14. Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning // Part I, Information Sciences. 1975. № 8(3). pp. 199–249.

15. Garg H., Dutta, D., Dutta P., Gohain B. An extended group decision-making algorithm with intuitionistic fuzzy set information distance measures and their applications // *Computers & Industrial Engineering*. 2024. 197. DOI: 10.1016/j.cie.2024.110537.
16. Seth T., Muhuri P.K. Hesitant and uncertain linguistics based executive decision making using risk and regret aversion: Methods, implementation and analysis // *MethodsX*. 2024. vol. 12.
17. Bozhenyuk A., Gerasimenko E., Kacprzyk J., Rozenberg I. Flows in networks under fuzzy conditions // *Studies in Fuzziness and Soft Computing*. 2017. vol. 346. 168 p.
18. Gerasimenko E., Kureichik V., Kuliev E. Maximum Dynamic Flow Model for Hesitant Fuzzy Evacuation with Intermediate Storage at Nodes // *Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation (INFUS)*. *Lecture Notes in Networks and Systems*. 2022. vol. 307. pp. 694–704.
19. Gerasimenko E., Kureichik V. Minimum cost lexicographic evacuation flow finding in intuitionistic fuzzy networks // *Journal of Intelligent and Fuzzy Systems*. 2022. vol. 42(1). pp. 251–263.
20. Герасименко Е.М. Курейчик В.В. Родзин С.И. Кухаренко А.П. Применение нечеткой логики для принятия решений об эвакуации при наводнении // *Известия ЮФУ. Технические науки*. 2022. № 4(228). С. 15–29.
21. Гамаюнова В.О., Богомолов А.С., Кушников В.А., Иващенко В.А. Мультиагентное моделирование эвакуации из помещений с учетом столкновений агентов // *Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика*. 2025. Т. 25. № 1. С. 106–115.
22. Коткова Е.А., Матвеев А.В. Методика интеллектуального прогнозирования эффективности управления эвакуацией людей из общественных зданий // *Научно-аналитический журнал "Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России"*. 2021. № 4. С. 107–120.
23. Ashraf S., Garg H., Kousar M. An industrial disaster emergency decision-making based on China's Tianjin city port explosion under complex probabilistic hesitant fuzzy soft environment. *Engineering Applications of Artificial Intelligence*. 2023. vol. 123. DOI: 10.1016/j.engappai.2023.106400.
24. Luo Z., Tian J., Zeng J., Pilla F. Flood risk evaluation of the coastal city by the EWM-TOPSIS and machine learning hybrid method. *International Journal of Disaster Risk Reduction*. 2024. vol. 106.
25. Dhamala M.C., Adhikari D., Khanal P. Pyakurel U. Generalized maximum flow over time with intermediate storage // *Annals of Operations Research*. 2024. vol. 335(1). pp. 111–134.
26. Xu Z. Hesitant Fuzzy Sets Theory // *Studies in Fuzziness and Soft Computing*. 2014. vol. 314.
27. D'Urso P., Chachi J., Kazemifard A., De Giovanni L. OWA-based multi-criteria decision making based on fuzzy methods // *Ann. Oper. Res.* 2024. DOI: 10.1007/s10479-024-05926-5.
28. Kazemifard A., Chachi J. MADM approach to analyse the performance of fuzzy regression models. *Journal of Ambient Intelligence and Humanized Computing*. 2022. vol. 13(8). pp. 4019–4031.
29. Chen S.M., Yang M.W., Lee L.W., Yang S.W. Fuzzy multiple attributes group decision making based on ranking interval type-2 fuzzy sets // *Expert Systems with Applications*. 2012. vol. 39(5). pp. 5295–5308.
30. Wang W.Z., Liu X.W., Qin Y. Multi-attribute group decision making models under interval type-2 fuzzy environment // *Knowledge-Based Systems*. 2012. vol. 30. pp. 121–128.

31. Zadeh L.A. Fuzzy sets // *Information and Control*. 1965. vol. 8. pp. 338–353.
32. Piegat A. Fuzzy Modelling and Control. Berlin Heidelberg. Springer-Verlag. 2001. 728 p.
33. Atanassov K.T. Intuitionistic fuzzy sets // *Fuzzy Sets and Systems*. 1986. vol. 20(1). pp. 87–96. DOI: 10.1016/S0165-0114(86)80034-3.
34. Ngan S.-C. An extension framework for creating operators and functions for intuitionistic fuzzy sets // *Information Sciences*. 2024. vol. 666.
35. Senapati T., Chen G., Mesiar R., Yager R.R. Intuitionistic fuzzy geometric aggregation operators in the framework of Aczel-Alsina triangular norms and their application to multiple attribute decision making // *Expert Systems with Applications*. 2023. vol. 212. DOI: 10.1016/j.eswa.2022.118832.
36. Xu Z.S. Intuitionistic fuzzy aggregation operators // *IEEE Transactions on Fuzzy Systems*. 2007. vol. 15(6). pp. 1179–1187.
37. Ye J. Group decision-making strategy based on aggregation operators of linguistic confidence interval neutrosophic numbers in a linguistic neutrosophic multivalued scenario // *Engineering Applications of Artificial Intelligence*. 2025. vol. 141. DOI: 10.1016/j.engappai.2024.109823.
38. Faizi S., Shah M.H., Sahabun W., Rashid T. A novel approach employing hesitant intuitionistic fuzzy linguistic Einstein aggregation operators within the EDAS approach for multicriteria group decision making // *Heliyon*. 2024. vol. 10(11).
39. Faizi S., Shah M., Rashid T. A modified VIKOR method for group decision-making based on aggregation operators for hesitant intuitionistic fuzzy linguistic term sets // *Soft Computing*. 2022. vol. 26(5). pp. 2375–2390.
40. Torra V. The weighted OWA operator // *International Journal of Intelligent Systems*. 1997. vol. 12(2). pp. 153–166.
41. Kacprzyk J., Bozhenyuk A., Gerasimenko E. Lexicographic maximum dynamic evacuation modeling with partial lane reversal based on hesitant fuzzy TOPSIS // *Applied Soft Computing*. 2023. vol. 144. DOI: 10.1016/j.asoc.2023.110435.
42. Gerasimenko E. Dynamic Flow Algorithm with Intermediate Storage for Emergency Evacuation Allowing Lane Reversal based on Incomplete Intuitionistic Fuzzy Preference Relation // *The open transportation journal*. 2024. vol. 18(1). DOI: 10.2174/0118744478281682240627062300.
43. Khanal D.P., Pyakurel U., Dhamala T.N., Dempe S., Schiermeyer I. Prioritized Maximum Multi-Commodity Flow in Evacuation Planning Dynamics of Disasters // *Natural Phenomena to Human Activity*. 2024. pp. 123–140.
44. Adhikari M.C., Pyakurel U. Lexicographic Maximum Flow Allowing Intermediate Storage // *The Nepali Mathematical Sciences Report*. 2022. vol. 39. no. 1. pp. 1–13.
45. Wang L., Gong Z., Zhang N. Consensus Modelling on Interval-Valued Fuzzy Preference Relations with Normal Distribution // *International Journal of Computational Intelligence Systems*. 2018. vol. 11(1). pp. 706–715.
46. Chen K.-S., Hsieh T.-H., Chang C.-P., Yao K.-C., Huang T.-H. Fuzzy Decision-Making and Resource Management Model of Performance Evaluation Indices // *Axioms*. 2024. vol. 13(3).
47. Baral S., Parida P., Sahoo D. Fuzzy TOPSIS technique for multi-criteria group decision-making: A study of crude oil price // *Results in Control and Optimization*. 2025. vol. 19. DOI: 10.1016/j.rico.2025.100565.
48. Герасименко Е.М., Кравченко Д.Ю., Курейчик В.В., Кулиев Э.В., Кравченко Ю.А., Родзин С.И. Модифицированный биоинспирированный метод поддержки принятия решений по предупреждению и ликвидации последствий чрезвычайных ситуаций // *Информационные технологии*. 2023. Т. 29. № 8. С. 423–436. DOI: 10.17587/it.29.423-436.

49. Buckley J.J. The Fuzzy Mathematics of Finance // Fuzzy Sets and Systems. 1987. № 21(3). pp. 257–273.

Курейчик Владимир Викторович — д-р техн. наук, профессор, заведующий кафедрой, кафедра САПР им. В.М. Курейчика, ИКТИБ, Южный Федеральный университет. Область научных интересов: биоинспирированные алгоритмы, эволюционные вычисления. Число научных публикаций — 500. vkur@sfedu.ru; улица Энгельса, 1, 347928, Таганрог, Россия; р.т.: +7(863)438-3451.

Герасименко Евгения Михайловна — канд. техн. наук, доцент кафедры, кафедра САПР им. В.М. Курейчика, ИКТИБ, Южный Федеральный университет. Область научных интересов: нечеткая логика, принятие решений в нечеткой среде, потоковые алгоритмы, эвакуационные задачи. Число научных публикаций — 110. egerasimenko@sfedu.ru; улица Энгельса, 1, 347928, Таганрог, Россия; р.т.: +7(863) 437-1651.

Поддержка исследований. Исследование выполнено за счет гранта Российского научного фонда № 22-71-10121-П, <https://rscf.ru/project/22-71-10121-П/> в Южном федеральном университете.

V. KUREICHIK, E. GERASIMENKO
**INTEGRATED ALGORITHM FOR MULTI-CRITERIA GROUP
DECISION-MAKING IN EVACUATION TASKS**

Kureichik V., Gerasimenko E. **Integrated Algorithm for Multi-Criteria Group Decision-Making in Evacuation Tasks.**

Abstract. This article presents an integrated algorithm for multicriteria group decision-making based on an intuitionistic fuzzy hybrid averaging operator for selecting the optimal evacuation strategy. The algorithm is multilevel, where the first level involves constructing a task model in the form of an evacuation transportation network. At the second level, the ranking of evacuation spaces is performed to determine their optimal order based on various criteria, such as room capacity, ease of evacuation, safety level of the room as a shelter, time required to organize evacuation from the location, and distance from the source of danger. The ranking of evacuation spaces is carried out using multicriteria group decision-making and an intuitionistic fuzzy hybrid averaging operator to model the doubts and uncertainties of experts in evaluating evacuation criteria, alternatives, and expert importance. The algorithm operates with linguistic expert assessments, allowing for the calculation of expert and criterion weights based on these assessments for effective decision-making. Aggregation of assessments is performed using a modified algorithm that allows for the operation of criterion weights represented as intuitionistic fuzzy values, unlike traditional crisp numbers, based on developed modified operators for raising fuzzy numbers to a fuzzy power to account for the doubts and uncertainties of the expert group. At the third level, macroscopic dynamic flow evacuation is carried out, considering the possibility of accommodating evacuees in spaces that are not shelters. The advantage of the proposed algorithm is its ability to model the transportation of evacuees under dynamic conditions from hazardous areas, taking into account their placement in intermediate points to maximize the number of lives saved, considering the uncertainty of the environment, the fuzzy nature of factors influencing evacuation decisions, and the uncertainties and doubts of the expert group in evaluating evacuation strategies. To confirm the effectiveness of the developed algorithm, evacuation modeling was conducted, and a software environment was created, implemented in JavaScript. A comparison of the developed decision-making algorithm based on the IFHA operator, operating with fuzzy weights of criteria, with existing algorithms was carried out, and a conclusion was made about the reliability of the proposed algorithm. An assessment of the dependence of the algorithm's runtime on the input size was conducted, confirming the possibility of using the proposed algorithm for large buildings and transportation networks.

Keywords: fuzzy intuitionistic decision making, fuzzy macroscopic evacuation, fuzzy aggregation operators.

References

1. Gerasimenko E.M., Kravchenko D.Ju., Kravchenko Ju.A., Kuliev Je.V. [Decision-making support for the prevention and elimination of consequences of emergency situations based on a fuzzy method of structuring information]. *Izvestiya SFU. Technical sciences – Izvestija JuFU. Tehnicheskie nauki*. 2023. no. 2(232). pp. 201–212. (In Russ.).
2. Shi D., Li J., Wang Q., Chen J., Lovreglio R., Lo J.T.Y., Ma J. Modeling and simulation of crowd dynamics at evacuation bottlenecks during flood disasters. *Transportation Research Part E: Logistics and Transportation Review*. 2025. vol. 198.

3. Feng Z., Wang C., An J., Zhang X., Liu X., Ji X., Kang L., Quan W. Emergency fire escape path planning model based on improved DDPG algorithm. *Journal of Building Engineering*. 2024. vol. 95. DOI: 10.1016/j.jobe.2024.110090.
4. Kim J., Kim B.-J., Kim N. Perception-based analytical technique of evacuation behavior under radiological emergency: An illustration of the Kori area. *Nuclear Engineering and Technology*. 2021. № 53(3). pp. 825–832. DOI: 10.1016/j.net.2020.08.012.
5. Ford J., Fulkerson D.R. Maximal flow through a network. *Canadian Journal of Mathematics*. 1956. № 8(3). pp. 399–404.
6. Gale D. Transient flows in networks. *Michigan Mathematical Journal*. 1959. № 6(1). pp. 59–63.
7. Minieka E. Maximal, lexicographic, and dynamic network flows. *Oper. Res.* 1973. № 21(2). pp. 517–527. DOI: 10.1287/opre.21.2.517.
8. Chalmet L.G., Francis R.L., Saunders P.B. Network models for building evacuation. *Fire Technology*. 1982. № 18(1). pp. 90–113. DOI: 10.1007/BF02993491.
9. Opananon S. On finding paths and flows in multicriteria, stochastic and time-varying networks. Ph.D. dissertation. University of Maryland. 2004.
10. Cui J., An S., Zhao M. A Generalized Minimum Cost Flow Model for Multiple Emergency Flow Routing. *Mathematical Problems in Engineering*. 2014. DOI: 10.1155/2014/832053.
11. Hamacher H.W., Tufekci S. On the use of lexicographic min cost flows in evacuation modeling. *Naval Research Logistics*. 1987. vol. 34(4). pp. 487–503.
12. Dunn C.E., Newton D. Optimal routes in GIS and emergency planning applications. *Area*. 1992. vol. 24(3). pp. 259–267.
13. Miller-Hooks E., Patterson S.S. On solving quickest time problems in time-dependent, dynamic networks. *Journal of Mathematical Modelling and Algorithms*. 2004. № 3(1). pp. 39–71.
14. Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning. Part I. *Information Sciences*. 1975. № 8(3). pp. 199–249.
15. Garg H., Dutta, D., Dutta P., Gohain B. An extended group decision-making algorithm with intuitionistic fuzzy set information distance measures and their applications. *Computers & Industrial Engineering*. 2024. 197. DOI: 10.1016/j.cie.2024.110537.
16. Seth T., Muhuri P.K. Hesitant and uncertain linguistics based executive decision making using risk and regret aversion: Methods, implementation and analysis. *MethodsX*. 2024. vol. 12.
17. Bozhenyuk A., Gerasimenko E., Kacprzyk J., Rozenberg I. Flows in networks under fuzzy conditions. *Studies in Fuzziness and Soft Computing*. 2017. vol. 346. 168 p.
18. Gerasimenko E., Kureichik V., Kuliev E. Maximum Dynamic Flow Model for Hesitant Fuzzy Evacuation with Intermediate Storage at Nodes. *Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation (INFUS)*. *Lecture Notes in Networks and Systems*. 2022. vol. 307. pp. 694–704.
19. Gerasimenko E., Kureichik V. Minimum cost lexicographic evacuation flow finding in intuitionistic fuzzy networks. *Journal of Intelligent and Fuzzy Systems*. 2022. vol. 42(1). pp. 251–263.
20. Gerasimenko E.M. Kureichik V.V. Rodzin S.I. Kuharenko A.P. [Applying fuzzy logic to make decisions about flood evacuation]. *Izvestija JuFU. Tehnicheskie nauki – Izvestiya SFU. Technical sciences*. 2022. no. 4(228). pp. 15–29. (In Russ.).
21. Gamayunova V.O., Bogomolov A.S., Kushnikov V.A., Ivashchenko V.A. Multi-agent simulation of evacuation from premises, taking into account agent collisions. *Izvestija Saratovskogo universiteta. Novaja serija. Serija: Matematika. Mehanika. Informatika*

- Proceedings of the Saratov University. A new series. Series: Mathematics. Mechanics. Computer science. 2025. vol. 25. no. 1. pp. 106–115. (In Russ.).
22. Kotkova E.A., Matveev A.V. [The methodology of intelligent forecasting of the effectiveness of management of evacuation of people from public buildings]. *Nauchno-analiticheskij zhurnal «Vestnik Sankt-Peterburgskogo universiteta Gosudarstvennoj protivopozharnoj sluzhby MChS Rossii»* – Scientific and analytical journal «Bulletin of the Saint Petersburg University of the State Fire Service of the Ministry of Emergency Situations of Russia». 2021. no. 4. pp. 107–120. (In Russ.).
23. Ashraf S., Garg H., Kousar M. An industrial disaster emergency decision-making based on China's Tianjin city port explosion under complex probabilistic hesitant fuzzy soft environment. *Engineering Applications of Artificial Intelligence*. 2023. vol. 123. DOI: 10.1016/j.engappai.2023.106400.
24. Luo Z., Tian J., Zeng J., Pilla F. Flood risk evaluation of the coastal city by the EWM-TOPSIS and machine learning hybrid method. *International Journal of Disaster Risk Reduction*. 2024. vol. 106.
25. Dhamala M.C., Adhikari D., Khanal P. Pyakurel U. Generalized maximum flow over time with intermediate storage. *Annals of Operations Research*. 2024. vol. 335(1). pp. 111–134.
26. Xu Z. *Hesitant Fuzzy Sets Theory*. Studies in Fuzziness and Soft Computing. 2014. vol. 314.
27. D'Urso P., Chachi J., Kazemifard A., De Giovanni L. OWA-based multi-criteria decision making based on fuzzy methods. *Ann. Oper. Res.* 2024. DOI: 10.1007/s10479-024-05926-5.
28. Kazemifard A., Chachi J. MADM approach to analyse the performance of fuzzy regression models. *Journal of Ambient Intelligence and Humanized Computing*. 2022. vol. 13(8). pp. 4019–4031.
29. Chen S.M., Yang M.W., Lee L.W., Yang S.W. Fuzzy multiple attributes group decision making based on ranking interval type-2 fuzzy sets. *Expert Systems with Applications*. 2012. vol. 39(5). pp. 5295–5308.
30. Wang W.Z., Liu X.W., Qin Y. Multi-attribute group decision making models under interval type-2 fuzzy environment. *Knowledge-Based Systems*. 2012. vol. 30. pp. 121–128.
31. Zadeh L.A. Fuzzy sets. *Information and Control*. 1965. vol. 8. pp. 338–353.
32. Piegat A. *Fuzzy Modelling and Control*. Berlin Heidelberg. Springer-Verlag. 2001. 728 p.
33. Atanassov K.T. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*. 1986. vol. 20(1). pp. 87–96. DOI: 10.1016/S0165-0114(86)80034-3.
34. Ngan S.-C. An extension framework for creating operators and functions for intuitionistic fuzzy sets. *Information Sciences*. 2024. vol. 666.
35. Senapati T., Chen G., Mesiar R., Yager R.R. Intuitionistic fuzzy geometric aggregation operators in the framework of Aczel-Alsina triangular norms and their application to multiple attribute decision making. *Expert Systems with Applications*. 2023. vol. 212. DOI: 10.1016/j.eswa.2022.118832.
36. Xu Z.S. Intuitionistic fuzzy aggregation operators. *IEEE Transactions on Fuzzy Systems*. 2007. vol. 15(6). pp. 1179–1187.
37. Ye J. Group decision-making strategy based on aggregation operators of linguistic confidence interval neutrosophic numbers in a linguistic neutrosophic multivalued scenario. *Engineering Applications of Artificial Intelligence*. 2025. vol. 141. DOI: 10.1016/j.engappai.2024.109823.
38. Faizi S., Shah M.H., Safabun W., Rashid T. A novel approach employing hesitant intuitionistic fuzzy linguistic Einstein aggregation operators within the EDAS approach for multicriteria group decision making. *Heliyon*. 2024. vol. 10(11).

39. Faizi S., Shah M., Rashid T. A modified VIKOR method for group decision-making based on aggregation operators for hesitant intuitionistic fuzzy linguistic term sets. *Soft Computing*. 2022. vol. 26(5). pp. 2375–2390.
40. Torra V. The weighted OWA operator. *International Journal of Intelligent Systems*. 1997. vol. 12(2). pp. 153–166.
41. Kacprzyk J., Bozhenyuk A., Gerasimenko E. Lexicographic maximum dynamic evacuation modeling with partial lane reversal based on hesitant fuzzy TOPSIS. *Applied Soft Computing*. 2023. vol. 144. DOI: 10.1016/j.asoc.2023.110435.
42. Gerasimenko E. Dynamic Flow Algorithm with Intermediate Storage for Emergency Evacuation Allowing Lane Reversal based on Incomplete Intuitionistic Fuzzy Preference Relation. *The open transportation journal*. 2024. vol. 18(1). DOI: 10.2174/0118744478281682240627062300.
43. Khanal D.P., Pyakurel U., Dhamala T.N., Dempe S., Schiermeyer I. Prioritized Maximum Multi-Commodity Flow in Evacuation Planning Dynamics of Disasters. *Natural Phenomena to Human Activity*. 2024. pp. 123–140.
44. Adhikari M.C., Pyakurel U. Lexicographic Maximum Flow Allowing Intermediate Storage. *The Nepali Mathematical Sciences Report*. 2022. vol. 39. no. 1. pp. 1–13.
45. Wang L., Gong Z., Zhang N. Consensus Modelling on Interval-Valued Fuzzy Preference Relations with Normal Distribution. *International Journal of Computational Intelligence Systems*. 2018. vol. 11(1). pp. 706–715.
46. Chen K.-S., Hsieh T.-H., Chang C.-P., Yao K.-C., Huang T.-H. Fuzzy Decision-Making and Resource Management Model of Performance Evaluation Indices. *Axioms*. 2024. vol. 13(3).
47. Baral S., Parida P., Sahoo D. Fuzzy TOPSIS technique for multi-criteria group decision-making: A study of crude oil price. *Results in Control and Optimization*. 2025. vol. 19. DOI: 10.1016/j.rico.2025.100565.
48. Gerasimenko E.M., Kravchenko D.Ju., Kureychik V.V., Kuliev Je.V., Kravchenko Ju.A. Ju.A., Rodzin S.I. [A modified bioinspired decision support method for the prevention and elimination of consequences of emergency situations]. *Informacionnye tehnologii – Information technology*. 2023. vol. 29. no. 8. pp. 423–436. DOI: 10.17587/it.29.423-436. (In Russ.).
49. Buckley J.J. *The Fuzzy Mathematics of Finance. Fuzzy Sets and Systems*. 1987. № 21(3). pp. 257–273.

Kureichik Vladimir — Ph.D., Dr.Sci., Professor, Head of the department, V.M. Kureychik CAD department, ICTIS, Southern Federal University. Research interests: bioinspired algorithms, evolutionary computation. The number of publications — 500. vkur@sfedu.ru; 1, Engels St., 347928, Taganrog, Russia; office phone: +7(863)438-3451.

Gerasimenko Evgeniya — Ph.D., Associate professor of the department, V.M. Kureychik CAD department, ICTIS, Southern Federal University. Research interests: fuzzy logic, decision-making in a fuzzy environment, flow algorithms, evacuation tasks. The number of publications — 110. egerasimenko@sfedu.ru; 1, Engels St., 347928, Taganrog, Russia; office phone: +7(863) 437-1651.

Acknowledgements. The research was funded by the Russian Science Foundation Project No. 22-71-10121-II, <https://rscf.ru/en/project/22-71-10121-II/> implemented by the Southern Federal University.

Е.В. ЛАРКИН, А.В. БОГОМОЛОВ, А.Н. ПРИВАЛОВ
**НЕЙРОСЕТЕВАЯ НАСТРОЙКА
ГЕНЕТИЧЕСКОГО АЛГОРИТМА УПРАВЛЕНИЯ
ВЕКТОРНОЙ ИЕРАРХИЧЕСКОЙ СИСТЕМОЙ**

Ларкин Е.В., Богомолов А.В., Привалов А.Н. Нейросетевая настройка генетического алгоритма управления векторной иерархической системой.

Аннотация. Представлены результаты исследования иерархической двухуровневой системы векторного управления многосвязным объектом, эволюция которого описывается вектором состояния, изменяющегося путем воздействия на исполнительные устройства, каждое из которых включает привод и рабочий механизм. Рассматриваемая система управления отличается наличием дополнительного контура настройки виртуального регулятора на верхнем и функционально-логическом уровнях. Синтезирована математическая модель импульсного отклика исполнительного механизма системы, учитывающая сухое трение, люфт и ограничения по скорости и положению рабочего органа управляемого объекта. Исходная модель исполнительного органа представлена в форме Коши, а его импульсный отклик аппроксимирован импульсным откликом линейного звена второго порядка, оптимальным по критерию минимума ошибки аппроксимации. Доказано, что параметры линеаризованного импульсного отклика зависят от эксплуатационных параметров привода. Построена модель замкнутой системы управления объектом в целом и показано, что ее параметры зависят от эксплуатационных параметров приводов, желаемого значения вектора состояния объекта управления и параметров виртуальных регуляторов, реализованных на функционально-логическом и верхнем иерархическом уровнях. Полученные результаты демонстрируют, что изменение эксплуатационных параметров объекта может быть скомпенсировано за счет структурных и параметрических изменений генетического алгоритма управления. Разработана методика синтеза генетического алгоритма управления сложными многоконтурными объектами, реализуемого контроллером верхнего уровня иерархии, базирующаяся на использовании нейронной сети. Показано, что предлагаемый подход обеспечивает достижение синергетического эффекта, когда управляющие воздействия, реализуемые разными модификациями алгоритма управления, оказываются менее эффективными, чем управляющие воздействия, реализуемые с помощью составного алгоритма, подвергающегося эволюционным изменениям в процессе функционирования системы. Корректность теоретических положений подтверждена результатами вычислительного моделирования управления виртуальным регулятором с помощью нейронной сети, продемонстрировавшего существенное улучшение характеристик управления за счет уменьшения времени выхода на установившийся режим и времени перерегулирования.

Ключевые слова: иерархическая система управления, векторное управление, многосвязный объект, генетический алгоритм, нейронная сеть, виртуальный регулятор.

1. Введение. Использование универсальных контроллеров в качестве аппаратной платформы контроля технологических процессов промышленных объектов существенно расширяет класс законов управления, которые могут быть реализованы в системе [1 – 3]. Принцип гибкого программирования, реализованный в приборах подобного типа, позволяет, не меняя аппаратной части контроллера

управлять объектами, структура и параметры которых меняются в процессе функционирования в широких диапазонах, в том числе и такими, которые в некотором диапазоне регулируемых величин являются структурно неустойчивыми. К последним могут быть отнесены многосвязные объекты управления, описываемые вектором состояния, воздействие на которые осуществляется с помощью множества исполнительных устройств, каждое из которых включает привод и рабочий механизм [4, 5]. В местах стыковки привода и рабочего механизма исполнительного устройства всегда присутствуют существенные нелинейности типа люфт, сухое трение, ограничение по скорости и положению рабочего механизма [6, 7]. Параметры как собственно объекта, так и исполнительного устройства могут изменяться в процессе эксплуатации системы, причем вследствие наличия перекрестных связей между контурами изменение эксплуатационных параметров в каждом из каналов влияет на статические и динамические характеристики замкнутой системы в целом. Так, увеличение сухого трения в механизмах понижает точность поддержания регулируемых параметров, а увеличение люфта приводит к увеличению перерегулирования и времени выхода системы на установившиеся режимы. Для повышения эксплуатационной устойчивости систему управления на физическом уровне разделяют на верхний и функционально-логический уровни, что позволяет гибко изменять как алгоритмы управления контроллеров, замыкающих обратные связи по состоянию рабочих механизмов исполнительных устройств, так и алгоритма управления объектом в целом [8, 9].

Таким образом, сложность объекта управления, создаваемый избыток вычислительных мощностей за счет физического разделения контроллеров, а также возможность оперативного гибкого перепрограммирования контроллеров создает предпосылки для реализации генетических алгоритмов управления [10, 11]. Подобные алгоритмы за счет эволюционной (параметрической) и мутационной (структурной) трансформации могут адаптироваться в режиме «on line» [12 – 14] к текущим условиям эксплуатации системы и текущему состоянию, как приводов и рабочих механизмов, так и самого объекта с сохранением основных характеристик управления таких, как устойчивость, перерегулирование, точность, время достижения желаемых значений регулируемых параметров состояния [15, 16]. Рабочим инструментом настройки алгоритмов может быть нейронная сеть [17, 18].

Методы создания генетических алгоритмов управления распространены на практике недостаточно, что подтверждает актуальность исследований.

Цель проведенного исследования заключается в создании метода построения генетических алгоритмов иерархических векторных систем управления, включающих дополнительный контур контроля структуры и параметров виртуального регулятора, работающего по принципу нейронной сети.

Постановка решаемой задачи может быть представлена в следующем виде. Имеется объект управления, состояние которого описывается вектором регулируемых параметров и зависит от вектора воздействий, формируемых рабочими механизмами, подключенными через подсистему приводов к иерархическому цифровому регулятору, включающему контроллер верхнего уровня иерархии и контроллеры функционально-логического уровня, каждый из которых управляет своим приводом, и в свою очередь, управляется контроллером верхнего уровня. Для обеспечения эксплуатационной устойчивости в систему, наряду с контурами, реализующими принцип обратной связи по состоянию рабочего механизма и объекта управления, введены контуры настройки виртуальных регуляторов, которые выполняют компенсирующую подстройку параметров регуляторов при изменении условий функционирования системы. Для этого на этапе разработки программного обеспечения должны быть построены линеаризованные математические модели приводов с рабочими механизмами, сформирована структура нейронной сети, определены варьируемые параметры виртуальных регуляторов, а также разработан алгоритм изменения структуры и параметров виртуальных регуляторов по сигналам обратной связи в дополнительном контуре управления. Интеграция алгоритма вариации параметров в алгоритм управления дают генетический алгоритм управления, для которого должны быть определены условия эволюции (изменение параметров операторов управленческой компоненты интегрального алгоритма) и мутации (изменение структуры управленческой компоненты, в частности, добавление, исключение, замена операторов).

2. Структура двухуровневой системы управления. На структурной схеме двухуровневой цифровой системы управления (рисунок 1) [8, 9] показаны OUC (object under control) – многосвязный объект управления, MC (master controller) – контроллер верхнего уровня иерархии; FLC (functional-logical level controller) FLC_i , $1 \leq i \leq K$ – контроллеры функционально-логического уровня иерархии; K – количество регулируемых параметров состояния объекта.

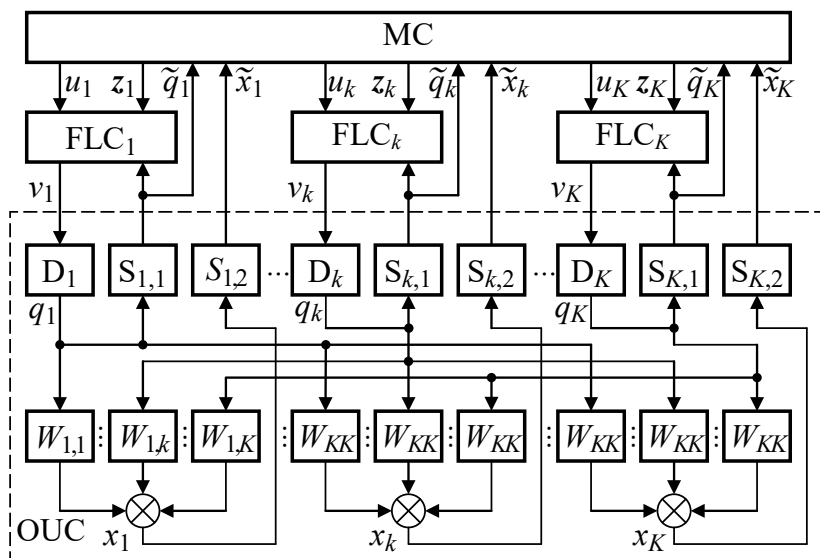


Рис. 1. Структурная схема двухуровневой системы управления

В объекте управления выделены физические исполнительные устройства D_k , включающие привод и рабочие механизмы (рули, заслонки, задвижки и т.п.); сенсоры $S_{k,1}$, измеряющие характеристики текущего состояния рабочих механизмов; сенсоры $S_{k,2}$, измеряющие параметры текущего состояния объекта управления.

Состояние объекта управления определяется вектором $\mathbf{x}(t) = [x_i(t)]^\theta$, где t – время; θ – обозначение операции транспонирования. Желаемые значения положений исполнительных механизмов рабочих органов задаются вектором $\mathbf{u}(t) = [u_i(t)]^\theta$, который генерируется контроллером верхнего уровня иерархии. Значения кодов вектора $\mathbf{u}(t)$ с выхода контроллера верхнего уровня иерархии подаются на управляющие входы контроллеров функционально-логического уровня иерархии. Контроллеры функционально-логического уровня иерархии формируют вектор сигналов $\mathbf{v}(t) = [v_i(t)]^\theta$, воздействующих на драйверы, в результате чего рабочие механизмы устанавливаются в состояния $\mathbf{q}(t) = [q_i(t)]^\theta$. Под воздействием $\mathbf{q}(t)$ объект управления переходит в состояние

$\mathbf{x}(t)$, параметры которого измеряются сенсорами $S_{2,1} - S_{2,K}$, вектор сигналов $\tilde{\mathbf{x}}(t) = [\tilde{x}_i(t)]^\theta$ вводится в контроллер верхнего уровня иерархии, в результате чего обратные связи по состоянию объекта управления в системе замыкаются.

Наряду с контурами, формируемыми для замыкания обратной связи по измеренному состоянию объекта $\tilde{\mathbf{x}}(t)$, в системе имеются контуры настройки виртуальных регуляторов, реализованных в контроллерах функционально-логического уровня иерархии. Настройка осуществляется за счет установки элементов вектора параметров $\mathbf{z}(t) = [z_i(t)]^\theta$. В свою очередь, параметры вектора $\mathbf{z}(t)$ формируются в соответствии с сигналами $\tilde{\mathbf{q}}(t) = [\tilde{q}_i(t)]^\theta$ сенсоров $S_{k,1}$ и сигналами $\tilde{\mathbf{x}}(t)$ сенсоров $S_{k,2}$.

3. Функционирование системы. Алгоритм работы контроллеров функционально-логического уровня иерархии при управлении исполнительными органами имеет следующий вид:

$$\mathbf{v}(t) = \mathbf{u}(t) - \mathbf{w}_c(t) * \tilde{\mathbf{q}}(t), \quad (1)$$

где $\mathbf{w}_c(t)$ – диагональная матрица размерности $K \times K$, описывающая закон управления, реализованный в контроллерах функционально-логического уровня иерархии; $*$ – матричная свертка, выполняемая по правилам перемножения матриц, в котором операция произведения заменена операцией свертки.

Элементы матрицы $\mathbf{w}_c(t) = [w_{c,k,l}(t)]$ описываются зависимостью:

$$w_{c,k,l}(t) = \begin{cases} 0, & \text{если } k \neq l; \\ w_{c,k}(t, \mathbf{z}_k), & \text{если } k = l; \end{cases} \quad (2)$$

где $w_{c,k,l}(t, \mathbf{z}_k)$ – импульсный отклик регулятора, реализуемый в контроллере программно; $\mathbf{z}_k$ – вектор параметров виртуального регулятора, устанавливаемых на контроллеры функционально-логического уровня иерархии.

Динамика исполнительных устройств системы определяется динамикой приводов рабочих механизмов и физических процессов, протекающих на стыке приводов и рабочих механизмов. В реальных драйверах, на стыке привода и рабочего механизма всегда присутствуют существенные нелинейности (рисунок 2), а именно, сухое трение (рисунок 2(а)), люфт с ограничением на перемещение (рисунок 2(б)) и ограничение по скорости (рисунок 2(в)) [6, 7].

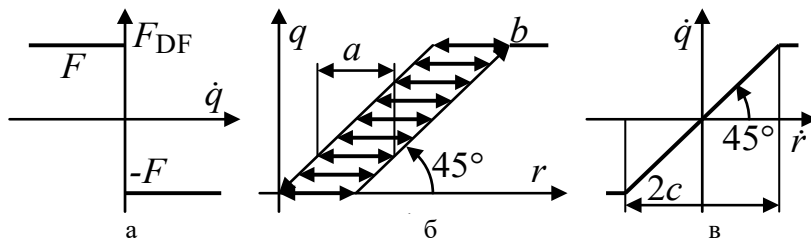


Рис. 2. Статические характеристики нелинейностей: а) сухое трение; б) люфт с ограничением; в) ограничение по скорости

В первом приближении сухое трение описывается выражением (для краткости индекс k опущен):

$$F_{DF} = -F \cdot \operatorname{sgn} \dot{q}, \quad (3)$$

где F – величина сухого трения; зависящая от нагрузки на трущиеся поверхности, рабочего механизма, а также от текущего состояния (загрязнение трущихся поверхностей, наличие/отсутствие смазки, и т.п.); $\dot{q}_k = dq_k/dt$. $\dot{q} = dq/dt$.

Статическая передаточная характеристика люфта $q(r)$ относится к неоднозначным передаточным характеристикам, зависящим от предыстории процесса, и ее конкретная реализация определяется не только текущей величиной аргумента $\tilde{q}$, но и направлением скорости изменения аргумента $\dot{\tilde{q}}$ в предыдущий момент времени $t - dt$, то есть

$$q(r, \dot{r}, t) = \begin{cases} 0, & \text{если } r(t) < 0; \\ r(t) - a, & \text{если } r(t) = q(t) + a \text{ и } \dot{r}(t - dt) > 0; \\ b, & \text{если } r(t) > a + b; \\ r, & \text{если } r(t) = q(t) \text{ и } \dot{r}(t - dt) < 0; \\ q(t - dt), & \text{если } r(t) = q(t) + a \text{ и } \dot{r}(t - dt) < 0, \\ q(t - dt), & \text{если } r(t) = q(t) \text{ и } \dot{r}(t - dt) > 0, \\ q(t - dt), & \text{если } q(t) < r(t) < q(t) + a, \end{cases} \quad (4)$$

где r – величина перемещения привода; q – величина перемещения рабочего механизма; b – верхнее ограничение на перемещение рабочего механизма; a – величина люфта.

Статическая передаточная характеристика, описывающая ограничение по скорости рабочего механизма, представляется следующим выражением:

$$\dot{q} = \begin{cases} \dot{r}, & \text{если } |\dot{r}| < c; \\ c, & \text{если } \dot{r} \geq c; \\ -c, & \text{если } \dot{r} \leq -c, \end{cases} \quad (5)$$

где c – предельное достигаемое значение скорости перемещения рабочего механизма.

С учетом существенных нелинейностей и особенностей работы привода может быть построена модель функционирования драйверов в форме Коши:

$$\begin{cases} \dot{r}_{k,1}(t) = r_{k,2}(t); \\ \dot{r}_{k,2}(t) = -\alpha_{k,2} r_{k,2}(t) + \alpha_{k,v} v_k(t) + \alpha_{k,R} R_k(t), \end{cases} \quad (6)$$

где $r_{k,1}(t) = r_k(t)$ – регулируемая координата k -го привода исполнительного устройства, например, угол поворота вала привода; $r_{k,2}(t)$ – скорость изменения регулируемой координаты привода; $[\alpha_{k,2}]^{-1}$ – постоянная времени разгонной характеристики привода;

$\alpha_{k,v}/\alpha_{k,2}$ – коэффициент передачи по управляющему воздействию;
 $\alpha_{k,R}/\alpha_{k,2}$ – коэффициент передачи по полезной нагрузке.

В соответствии с зависимостью (4) полезная нагрузка на валу привода появляется только когда $r(t) = q(t) + a$ и $\dot{r}(t - dt) > 0$, или когда $q(t) = r(t)$ и $\dot{r}(t - dt) < 0$. В этом случае

$$R_k(t) = \mu_k \dot{q}_{k,2}(t) + \eta_k q_{k,2}(t) + F_{DF,k}(t), \quad (7)$$

где $R_k(t)$ – нагрузка на валу привода; μ_k – приведенная масса рабочего механизма; η_k – приведенный коэффициент вязкого трения в подвижных частях рабочего механизма; $F_{DF,k}(t)$ – сила сухого трения, определяемая по зависимости (3).

Решение системы нелинейных уравнений (3) – (7) совместно с уравнением (1), описывающим замыкание обратной связи, при $v_k(t) = \delta(t)$, где $\delta(t)$ – δ -функция Дирака, $1 \leq k \leq K$, позволяет сформировать диагональную $K \times K$ матрицу импульсных откликов исполнительного устройства $\mathbf{w}_{DC}(t) = [w_{D,k,l}(t)]$. Элементы матрицы определяются как

$$w_{DC,k,l}(t) = \begin{cases} 0, & \text{если } k \neq l; \\ w_{DC,k}(t, \mathbf{z}_k, a_k, b_k, c_k, \eta_k, F_{DF,k}, \mu_k), & \end{cases} \quad (8)$$

где $\mathbf{z}_k$ – вектор параметров настройки k -го виртуального регулятора; a_k, b_k, c_k – значения параметров a, b, c существенных нелинейностей типа «люфт с ограничением по положению» и «ограничение по скорости» в k -м исполнительном устройстве; η_k и $F_{DF,k}$ – коэффициент вязкого трения и сила сухого трения в рабочем механизме исполнительного устройства; μ_k – приведенная масса исполнительного механизма.

Для упрощения решения задачи синтеза генетического алгоритма кривая импульсного отклика нелинейной замкнутой системы (8) должна быть аппроксимирована кривой отклика линейного звена второго порядка [8, 14, 15]:

$$\int_0^{\infty} \left\| w_{DC,k} \left(t, \mathbf{z}_k, a_k, b_k, c_k, \eta_k, F_{DF,k}, \mu_k \right) - \right. \\ \left. - \tilde{w}_{DC,k} \left(t, \kappa_{k,1}, \kappa_{k,2}, \kappa_{k,3} \right) \right\| dt \rightarrow \min, \quad (9)$$

где $w_{DC,k} \left(t, \mathbf{z}_k, a_k, b_k, c_k, \eta_k, F_{DF,k}, \mu_k \right)$ – аппроксимируемый импульсный отклик нелинейной модели; $\tilde{w}_{DC,k} \left(t, \kappa_{k,1}, \kappa_{k,2}, \kappa_{k,3} \right)$ – аппроксимирующий импульсный отклик линейной модели; $\kappa_{k,i} = \kappa_{k,i} \left(\mathbf{z}_k, a_k, b_k, c_k, \eta_k, F_{DF,k}, M_k \right)$, $i \in \{1, 2, 3\}$ – параметры оптимизации; $\|\dots\|$ – норма.

Линеаризация (9) позволяет представить описания замкнутых систем управления функционально-логического уровня с помощью аппарата передаточных функций:

$$\mathbf{Q}(s) = \tilde{\mathbf{W}}_{DC}(s) \cdot \mathbf{U}(s), \quad (10)$$

где $\mathbf{Q}(s) = L[\mathbf{q}(t)]$; $\mathbf{U}(s) = L[\mathbf{u}(t)]$; $\tilde{\mathbf{W}}_{DC}(s) = L[\tilde{\mathbf{w}}_{DC}(t)]$ $L[\dots]$ – прямое преобразование Лапласа; s – переменная Лапласа (оператор дифференцирования) [18, 19]; $\tilde{\mathbf{W}}_{DC}(s) = [\tilde{W}_{DC,k,l}(s)]$ – диагональная $K \times K$ матрица передаточных функций замкнутых систем управления исполнительными устройствами драйверов; $\tilde{W}_{DC,k,l}(s) = W_{DC,k}(s, \mathbf{z}_k)$ при $k=l$; $\mathbf{z}_k$ – вектор параметров, устанавливаемых на k -й виртуальный регулятор.

Для замыкания обратной связи через генетический алгоритм первичная математическая модель объекта управления, в общем случае нелинейная, должна быть линеаризована и преобразована в операторную форму, которая принимает вид:

$$\mathbf{X}(s) = \mathbf{W}_{OUC}(s, \{x_k^0\}) \cdot \mathbf{Q}(s), \quad 1 \leq k \leq K, \quad (11)$$

где $\mathbf{W}_{OUC}(s, \{x_k^0\}) = [W_{OUC,k,l}(s, \{x_k^0\})]$ – матрица передаточных функций самого общего вида размерности $K \times K$; $\{x_k^0\}$ – точка разложения первичной модели в ряд Тейлора; $\mathbf{X}(s) = L[\mathbf{x}(t)]$.

В общем случае в объекте управления имеются перекрестные связи между каналами формирования элементов вектора $\mathbf{X}(s)$ (рисунок 1), поэтому матрица $\mathbf{W}_{OUC}(s, \{x_i^0\})$ не является диагональной, и при $k \neq l$ не все ее элементы равны нулю. С учетом этого, для случая, если в генетическом алгоритме контроллера верхнего уровня иерархии реализован линейный закон управления, матричное уравнение, описывающее замкнутую систему управления, принимает вид:

$$\begin{aligned} \mathbf{X}(s) = & \left[\mathbf{E} + \mathbf{W}_{MC}(s, \{\mathbf{z}_{MC,k}\}) \cdot \mathbf{W}_{S,2}(s) \cdot \right. \\ & \left. \cdot \tilde{\mathbf{W}}_{DC}(s, \{\mathbf{z}_k\}) \cdot \mathbf{W}_{OUC}(s, \{x_i^0\}) \right]^{-1} \cdot \\ & \cdot \tilde{\mathbf{W}}_{DC}(s, \{\mathbf{z}_k\}) \cdot \mathbf{W}_{OUC}(s, \{x_i^0\}) \cdot \mathbf{F}(s), \end{aligned} \quad (12)$$

где $\mathbf{E}$ – единичная диагональная матрица; $\mathbf{W}_{MC}(s, \{\mathbf{z}_{MC,k}\})$ – диагональная матрица генетического алгоритма управления, реализованного в контроллере верхнего уровня иерархии; $\mathbf{z}_{MC,k} = [z_{MC,k,0(g)}, \{z_{MC,k,\gamma(g)}\}]$ – вектор описания k -го генетического алгоритма; $z_{MC,k,0(g)}$ – структура генетического алгоритма; $1(g,k) \leq \gamma(g,k) \leq \Gamma(g,k)$ – параметры; $\mathbf{W}_{S,2}(s)$ – диагональная матрица передаточных функций сенсоров $S_{2,i}$; $\mathbf{W}_{DC}(s, \mathbf{z}_i)$ – диагональная матрица передаточных функций исполнительных устройств; $\mathbf{F}(s) = L[\mathbf{f}(t)]$; $\mathbf{f}(t) = [f_k(t)]^0$ – вектор желаемых значений регулируемых параметров $\mathbf{x}(t)$, размерности всех диагональных матриц $K \times K$.

4. Настройка генетического алгоритма с помощью нейронной сети. Как следует из (9) и (11), статические и динамические характеристики замкнутой двухуровневой системы определяются эксплуатационными параметрами $a_i, b_i, c_i, \eta_i, F_{DF,i}, \mu_i$, $1 \leq i \leq K$, и значениями регулируемых параметров x_i^0 в точке разложения, зависящей от $\mathbf{f}(t)$. Перечисленные параметры могут значительно изменяться в процессе эксплуатации системы, однако эти изменения не должны оказывать влияния на ее работоспособность и такие характеристики, как перерегулирование, время выхода

на установившийся режим, точность. Изменения эксплуатационных параметров могут быть скомпенсированы на функционально-логическом уровне с помощью вариации векторов вектора $\mathbf{Z} = \mathbf{z} \cup \mathbf{z}_{MC}$. Настройка системы должна производиться по переходным процессам, наблюдаемым в драйверах D_k при замыкании обратных связей через FLC_k , а также по наблюдаемым переходным процессам в объекте управления при замыкании обратных связей через контроллер верхнего уровня иерархии.

Для управления настройкой наблюдаемые параметры представляются в виде обобщенного множества

$$\Xi(t) = \mathbf{f}(t) \cup \mathbf{u}(t) \cup \mathbf{v}(t) \cup \tilde{\mathbf{q}}(t) \cup \tilde{\mathbf{x}}(t), \quad (13)$$

где $\Xi(t) = \{ \xi_j(t) \}$, $1 \leq j \leq K$ – множество обобщенных наблюдаемых параметров системы.

Множество (13) описывает обобщенное пространство, в котором производится поиск эволюционно-мутационных вариаций генетических алгоритмов. Поиск параметров предлагается определять с помощью нейронной сети, структура которой показана на рисунке 3.

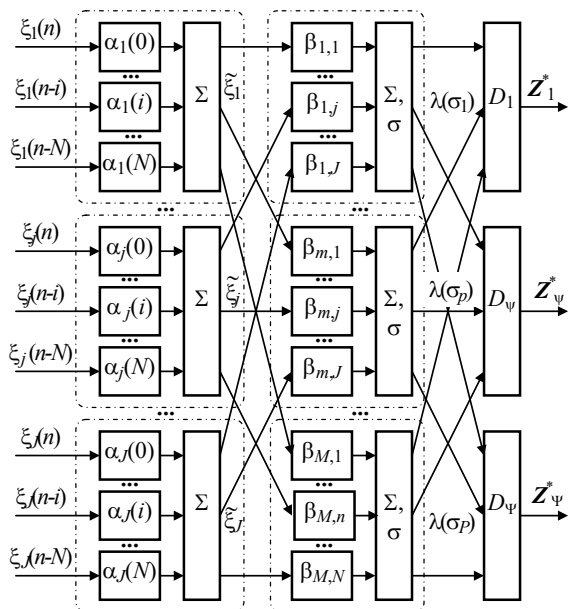


Рис. 3. Нейронная сеть управления параметрами регулятора

В сеть входят нейроны первого слоя, на входы которых подаются дискретные последовательности наблюдаемых сигналов

$$\xi_j(n), \dots, \xi_j(n-N), \quad (14)$$

где n – дискретный аналог физического времени, $N+1$ – длина последовательности.

Моменты физического времени $t_0, \dots, t_n, \dots, t_N$ и номера отсчетов $1, \dots, n, \dots, N$ связаны соотношением $t_n = 2\pi n\Omega^{-1}$, где Ω – круговая частота дискретизации, определяемая из условий Найквиста теоремы об отсчетах [8, 9].

Нейроны первого слоя осуществляют фильтрацию поступившей информации о состоянии драйверов D_k и объекта управления. Это может быть сделано с помощью вычисления линейной свертки [20, 21]

$$\tilde{\xi}_j(n) = \sum_{i=0}^N \alpha_j(i) \xi_j(n-i), \quad (15)$$

где $\alpha_j(i)$ – параметры фильтра, формируемые на этапе настройки нейронной сети.

Нейроны второго слоя формируют систему гиперплоскостей в J -мерном пространстве оценок параметров $\tilde{\xi}_j$

$$\beta \cdot \tilde{\xi} = y, \quad (16)$$

где $\beta = [\beta_{p,j}]$ – матрица параметров размерности $P \times J$; P – количество гиперплоскостей; J – размерность пространства; $y = (y_p)^{\theta}$ – вектор-столбец свободных членов, $1 \leq p \leq P$.

Матрица β и вектор y определяют расположение гиперплоскостей в пространстве параметров, которые необходимы для задания вектора логических σ -функций $\sigma = (\sigma_p)^{\theta}$, имеющих вид

$$\sigma_p = \begin{cases} 1, & \text{если } \sum_{j=1}^J \beta_{p,j} \xi_j > y_p; \\ 0, & \text{иначе,} \end{cases} \quad (17)$$

где 0, 1 – логический ноль и единица, соответственно.

Вследствие того, что каждая из σ -функций представляется логической переменной, она может быть представлена как аргумент логической функции

$$\lambda(\sigma_p) = \begin{cases} \sigma_p, & \text{если } \sigma_p = 1; \\ \bar{\sigma}_p, & \text{если } \sigma_p = 0, \end{cases} \quad (18)$$

где $\bar{}$ – логическая операция отрицания.

Логическая единица в (17) означает, что точка, заданная суммой

$$\sum_{j=1}^J \beta_{p,j} \xi_j, \text{ находится над плоскостью } \sum_{j=1}^J \beta_{p,j} \xi_j = y_p, \text{ а логический}$$

ноль – под указанной плоскостью. Таким образом, все пространство оценок параметров $\tilde{\xi}_j$ гиперплоскостями (16) разделяется на области, описываемые с помощью дизъюнктивной нормальной формы

$$D_\psi = \bigvee_i \left[\bigwedge_{p=1}^P \lambda_{\psi,i,p}(\sigma_p) \right], \quad (19)$$

где ψ – номер области; i – номер элементарной конъюнкции; p – номер гиперплоскости, ограничивающей область; $\lambda_{\psi,i,p}(\sigma_p)$ – логическая функция от σ_p при ее вхождении в i -ю элементарную конъюнкцию ψ -й области.

Каждая отдельная подобласть, описываемая в пространстве оценок состояния элементарной конъюнкцией, является выпуклой или открыто-выпуклой. Дизъюнктивные комбинации элементарных конъюнкций описывают объединения выпуклых подобластей, и позволяют сформировать области практически любого вида, в том числе открытые, невыпуклые и многосвязные. Множество подобластей, описанных совершенной дизъюнктивной нормальной формы покрывает все пространство наблюдаемых параметров (13). Количество подобластей определяется матричным уравнением (16), в частности зависит от матрицы коэффициентов β и вектора свободных членов y , которые задаются на этапе настройки нейронной сети. Эти параметры определяют вычислительную сложность процедуры коррекции генетического алгоритма. Увеличение количества подобластей повышает точность настройки, но

увеличивает также вычислительную сложность генетического алгоритма.

Для решения задачи управления параметрами каждой области D_ψ может быть поставлен в соответствие набор параметров

$$D_\psi \rightarrow \mathbf{Z}^* = \mathbf{z}_\psi^* \cup \mathbf{z}_{MC,\psi}^*, \quad (20)$$

где $\mathbf{z}_\psi^*$ – вектор векторов оптимальных параметров, устанавливаемых на виртуальные регуляторы контроллеров функционально-логического уровня иерархии, например ПИД регуляторов [22, 23]; $\mathbf{z}_{MC,\psi}^*$ – вектор текущих реализаций оптимальных генетических алгоритмов, реализованных в контроллере верхнего уровня иерархии.

Выбор (20) не накладывает никаких ограничений на алгоритмы управления, выбираемые для реализации. В том случае, если в каждом из выбираемых алгоритмов изменяется только параметры регуляторов, то эти изменения могут быть отнесены к эволюционным, а если изменяется принцип управления и параметры регулятора, то изменения могут быть отнесены к мутационным. Примером эволюционных вариаций может являться изменение коэффициентов передачи по пропорциональному, интегральному и дифференциальному каналам виртуального ПИД-регулятора. Примером мутационных изменений является случай, когда при большой ошибке в k -м канале

$$\|f_k(t) - x_k(t)\| > \varepsilon_k, \quad (21)$$

где ε_k – порог мутации, выбирается релейный способ управления, то есть $u_k = u_{k,\max} \operatorname{sgn}[f_k(t) - x_k(t)]$.

При малой ошибке, когда (21) не выполняется, закон управления может быть выбран линейным, например, в контур обратной связи этого канала может быть включен линейный ПИД регулятор. При этом релейный закон управления обеспечивает максимально быстрый выход параметра $x_k(t)$ из области (21), а параметры ПИД регулятора должны обеспечивать устойчивое удержание величины $x_k(t)$ вне зоны (21).

Таким образом, нейронная сеть формирует в системе дополнительный контур управления, в котором на основании анализа качественных характеристик переходных процессов в приводах

и системе в целом выбирается принцип управления объектом и параметры регулятора, обеспечивающие заданные показатели качества: устойчивость, перерегулирование, время выхода на установившийся режим, точность. Как следует из логики работы первого слоя нейронов для принятия решения о выборе Z^* необходимо, чтобы прошло как минимум, $J \cdot (N+1)$ транзакций по вводу данных о состоянии объекта. Таким образом, в системе на каждые $N+1$ циклов управления должен приходиться один цикл коррекции структуры и/или параметров генетического алгоритма.

5. Пример синтеза регулятора. Реализуемость этапов процесса синтеза генетического алгоритма показана на следующем примере.

В качестве исполнительного устройства в системе используется привод постоянного тока с управлением по якорной обмотке наиболее часто применяемый в системах управления промышленными объектами. В месте стыковки двигателя с рабочим механизмом действуют все существенные нелинейности, представленные на рисунке 2. Формирование эквивалентного линеаризованного импульсного отклика привода показано на рисунке 4(а).

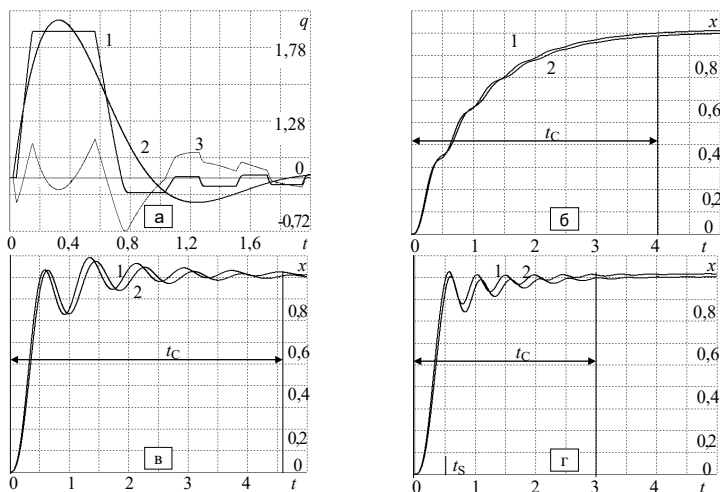


Рис. 4. Работа двухуровневой системы управления

Первичная модель привода имеет вид (1) – (8). Линейная часть привода с учетом электрической и механической постоянных времени

описывается передаточной функцией $W(s) = \frac{1,3}{0,01 \cdot s^2 + 0,4 \cdot s + 1}$.

Параметры нелинейной части модели являются следующими: $a = 0,2$, $b = 0,5$; $c = 0,1$; $F = 0,2$. Кривая 1 показывает импульсный отклик нелинейной системы. Кривая 2 показывает импульсный отклик линеаризованной системы, с передаточной функцией

$$W_{DC}(s) = \frac{1}{0,063 \cdot s^2 + 0,256 \cdot s + 1},$$

полученной как решение оптимизационной задачи (8). Кривая (3) показывает разность между импульсными откликами нелинейной и линеаризованной систем.

Объект управления описывается линейным матричным уравнением

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} \frac{5}{s+1} & \frac{0,4}{0,8 \cdot s + 1} \\ -\frac{0,4}{0,8 \cdot s + 1} & \frac{5}{s+1} \end{bmatrix} \cdot \begin{bmatrix} Q_1(s) \\ Q_2(s) \end{bmatrix},$$

где $X_1(s) = L[x_1(t)]$; $X_2(s) = L[x_2(t)]$; $Q_1(s) = L[q_1(t)]$; $Q_2(s) = L[q_2(t)]$.

В двухуровневой системе при управлении состоянием объекта управления использован виртуальный ПИД-регулятор [22, 23].

На рисунке 4(б) показан переходный процесс для случая, когда коэффициенты по пропорциональному и дифференциальному каналам виртуального регулятора равны $z_{MC,P,1} = z_{MC,P,2} = 1$, $z_{MC,D,1} = z_{MC,D,2} = 1$. Выход на установившийся режим осуществляется без перегулирования за время $t_c = 4$ с.

На рисунке 4(в) показан переходный процесс для случая, когда коэффициенты по пропорциональному и дифференциальному каналам равны $z_{MC,P,1} = z_{MC,P,2} = 1$, $z_{MC,D,1} = z_{MC,D,2} = 0,3$. Выход на установившийся режим осуществляется с перегуливанием 12% за время $t_c = 4,6$ с.

На рисунке 4(г) показан переходный процесс для случая, когда для управления применяется генетический алгоритм. Разделяющие поверхности выражения (16) имеют вид $\bar{\xi}_1 = x_1 = 0,8$ $\bar{\xi}_2 = x_2 = 0,8$. Если значения регулируемых параметров x_1, x_2 меньше 0,8, то контур настройки генетического алгоритма устанавливает значения $z_{MC,D,1} = z_{MC,D,2} = 0,3$, а если больше, то $z_{MC,D,1} = z_{MC,D,2} = 1$. Это происходит через 0,5 с после начала переходного процесса. При таком эволюционном изменении алгоритма время выхода на установившийся режим сокращается до 3 с, а перерегулирование не превышает 2%.

Таким образом, оперативная переналадка виртуального регулятора с помощью нейронной сети позволяет сократить время переходного процесса с четырех до 3 с, т.е. на 25% по сравнению со случаем рисунка 4(б), и с 4,6 до 3 с, т.е. на 34,7%, по сравнению со случаем рисунка 4(в). Синергетический эффект достигается за счет того, что в каждой конкретной области пространства состояний в иерархической системе управления действует версия алгоритма управления, обеспечивающая оптимальные режимы функционирования исполнительных устройств системы.

6. Заключение. В результате исследования разработана модель синтеза генетического алгоритма управления сложными многоконтурными объектами, реализуемого контроллером верхнего уровня иерархии, базирующуюся на применении нейронной сети. Показано, что с применением нейросетевой настройки генетического управляющего алгоритма достигается синергетический эффект, когда в целом параметры, достигаемые при управлении объектом с помощью разных модификаций алгоритма являются худшими по сравнению с параметрами, достигаемыми с помощью составного алгоритма, который подвергается эволюционным и/или мутационным изменениям в процессе функционирования системы.

Дальнейшее развитие работ в этом направлении связывается с разработкой методов рассечения пространства параметров на подобласти, в которых устанавливаются эволюционные и/или мутационные изменения генетического алгоритма управления, и обоснование оптимальной длительности временного интервала перенастройки генетического алгоритма.

Литература

1. Lofving M., Safsten K., Winroth M. Manufacturing strategy formulation, leadership style and organizational culture in small and medium-sized enterprises // International

- Journal of Manufacturing Technology and Management (IJMTM). 2016. vol. 30. no. 5. pp. 306–325. DOI: 10.1504/IJMTM.2016.078918.
2. Landau I.D., Zito G. Digital Control Systems, Design, Identification and Implementation. Springer, 2006. 484 p.
3. Astrom J., Wittenmark B. Computer Controlled Systems: Theory and Design. H.J.: Prentice Hall, 2002. 557 p.
4. Kilian C.T. Modern control technology: Components and systems. Novato, CA: Delmar Thomson Learning, 2000. 608 p.
5. Babishin V., Taghipour S. Optimal maintenance policy for multicomponent systems with periodic and opportunistic inspections and preventive replacements // Applied mathematical modelling. 2016. vol. 40. no. 23. pp. 10480–10505. DOI: 10.1016/j.apm.2016.07.019.
6. Акименко Т.А., Ларкин Е.В., Богомолов А.В., Привалов А.Н. Модель формирования образа сцены в телевизионном модуле мобильного робота, движущегося по пересеченной местности // Информационно-управляющие системы. 2024. № 5(132). С. 16–24. DOI: 10.31799/8853-2024-5-16-24.
7. Larkin E., Privalov A., Bogomolov A., Akimenko T. Digital control of continuous production with dry friction at actuators // Smart Innovation, Systems and Technologies. 2022. vol. 232. pp. 427–436.
8. Bogomolov A.V., Larkin E.V., Privalov A.N. Mathematical models of software failures of digital control systems // Automatic Documentation and Mathematical Linguistics. 2025. vol. 59. no. 1. pp. 1–8. DOI: 10.3103/S000510552470041.
9. Ларкин Е.В., Богомолов А.В., Привалов А.Н., Акименко Т.А. Математическое обеспечение иерархического цифрового управления сложным технологическим объектом // Вестник ЮрГУ. Серия. Математика. Механика. Физика. 2024. Т. 16. № 4. С. 43–55.
10. Harada T, Alba E. Parallel genetic algorithms: a useful survey // ACM Computing Survey. 2020. vol. 53. no. (4). pp. 1–39. DOI: 10.1145/3400031.
11. Hedar A., Fukushima M. Simplex coding genetic algorithm for the global optimization of nonlinear functions, in Multi-Objective Programming and Goal Programming // Advances in Soft Computing. Springer-Verlag, 2003. pp. 135–140. DOI: 10.1007/978-3-540-36510-5_17.
12. Helal M.H.S., Fan C.C., Liu D., Yuan S. Peer-to-peer based parallel genetic algorithm // International Conference on Information, Communication and Engineering (ICICE). Xiamen, 2017. pp. 535–538. DOI: 10.1109/ICICE.2017.8478917.
13. Hu C., Wang X., Mandal M.K., Meng M., Li D. Efficient face and gesture recognition techniques for robot control // Canadian Conference on Electrical and Computer Engineering. IEEE, Montreal, 2003. pp. 1757–1762. DOI: 10.1109/CCECE.2003.1226250.
14. Jebari K. Selection methods for genetic algorithms // International Journal of Emerging Sciences. 2013. no. 3. pp. 333–344.
15. Hamann A., Racu R., Ernst R. Multi-dimensional robustness optimization in heterogeneous distributed embedded systems // Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium, (RTAS'07). 2007. pp. 269–280. DOI: 10.1109/RTAS.2007.19.
16. Briat C. Stability and performance analysis of linear positive systems with delays using input-output methods // International Journal of Control. 2018. vol. 91. no. 7. pp. 1669–1692. DOI: 10.1080/00207179.2017.1326628.
17. Islam M., Chen G., Jin S. An overview of neural network // American Journal of Neural Network and Application. 2019. vol. 5. no. 1. pp. 7–11. DOI: 10.11648/j.ajjna.20190501.12.

18. Pavlov A.V. About the equality of the transform of Laplace to the transform of Fourier // *Issues of Analysis*. 2016. vol. 5. no. 4. pp. 21–30.
19. Li J., Farquharson C.G., Hu X. Three effective inverse Laplace transform algorithms for computing time -domain electromagnetic responses // *Geophysics*. 2015. vol. 81. no. 2. DOI: 10.1190/geo2015-0174.1.
20. Meyer-Baese U. *Digital signal processing*. Springer-Verlag Berlin, Heidelberg, 2004. 527 p.
21. Yeh Y.-C., Chu Y., Chiou C.W. Improving the sampling resolution of periodic signals by using controlled sampling interval method // *Computers & Electrical Engineering*. 2014. vol. 40. no. 4. pp. 1064–1071. DOI: 10.1016/j.compeleceng.2013.12.002.
22. Ang K.N., Chong G., Li Y. PID control system analysis, design and technology // *IEEE Transactions of control systems technology*. 2005. vol. 13. no. 4. pp. 559–576. DOI: 10.1109/TCST.2005.847331.
23. O'Dwier A. PID compensation of time delay processes 1999-2002: a survey // *Proceedings of the American control conference*. 2003. pp. 1494–1499. DOI: 10.21427/dkx2-9503.

Ларкин Евгений Васильевич — д-р техн. наук, профессор, почетный работник высшего профессионального образования РФ, профессор-консультант кафедры, кафедра «промышленная автоматика и робототехника», Тульский государственный университет. Область научных интересов: теория устойчивости иерархических многоконтурных систем управления сложными технологическими объектами. Число научных публикаций — 350. elarkin@mail.ru; проспект Ленина, 92, 300012, Тула, Россия; р.т.: +7(487)273-4444.

Богомолов Алексей Валерьевич — д-р техн. наук, профессор, ведущий научный сотрудник, Федеральный исследовательский центр «Информатика и управление» Российской академии наук. Область научных интересов: цифровые технологии сбора и обработки информации. Число научных публикаций — 350. a.v.bogomolov@gmail.com; улица Вавилова, 44/2, 119333, Москва, Россия; р.т.: +7(499)135-6260.

Привалов Александр Николаевич — д-р техн. наук, профессор, директор института, институт передовых информационных технологий, Тульский государственный педагогический университет им. Л.Н. Толстого. Область научных интересов: методы проектирования программного обеспечения информационных систем, тренажерных систем, компьютерное моделирование, распределённые вычисления. Число научных публикаций — 350. privaloov.61@mail.ru; проспект Ленина, 125а, 300026, Тула, Россия; р.т.: +7(487)235-1488.

Поддержка исследований. Работа выполнена при финансовой поддержке РНФ (проект № 24-29-20256).

E. LARKIN, A. BOGOMOLOV, A. PRIVALOV
**NEURAL NETWORK TUNING OF THE GENETIC ALGORITHM
FOR CONTROLLING A VECTOR HIERARCHICAL SYSTEM**

Larkin E., Bogomolov A., Privalov A. Neural Network Tuning of the Genetic Algorithm for Controlling a Vector Hierarchical System.

Abstract. The article presents the results of a study of a hierarchical two-level vector control system for a multi-connected object whose evolution is described by a state vector that changes in response to actions on actuators, each of which includes a drive and a working mechanism. The control system under consideration is distinguished by the presence of an additional tuning loop for the virtual regulator at the upper and functional-logical levels. A mathematical model of the impulse response of the actuator of the system is synthesized, taking into account dry friction, backlash, and limitations on the speed and position of the working element of the controlled object. The original model of the actuator is presented in the Cauchy form, and its impulse response is approximated by the impulse response of a second-order linear link, optimal according to the criterion of the minimum approximation error. It is proved that the parameters of the linearized impulse response depend on the operating parameters of the drive. A model of a closed control system for the object as a whole is constructed, and it is shown that its parameters depend on the operating parameters of the drives, the desired value of the state vector of the control object, and the parameters of the virtual controllers implemented at the functional-logical and upper hierarchical levels. The obtained results demonstrate that a change in the operating parameters of the object can be compensated for by structural and parametric changes in the genetic control algorithm. A technique for synthesizing a genetic control algorithm for complex multi-loop objects implemented by a controller at the upper level of the hierarchy based on the use of a neural network has been developed. It is shown that the proposed approach ensures the achievement of a synergetic effect, when the control actions implemented by different modifications of the control algorithm are less effective than the control actions implemented using a composite algorithm subject to evolutionary changes during the operation of the system. The correctness of the theoretical positions is confirmed by the results of computational modeling of the virtual controller control using a neural network, which demonstrated a significant improvement in the control characteristics due to a decrease in the time to reach a steady state and the overshoot time.

Keywords: hierarchical control system, vector control, multi-connected object, genetic algorithm, neural network, virtual controller.

References

1. Lofving M., Säfsen K., Winroth M. Manufacturing strategy formulation, leadership style and organizational culture in small and medium-sized enterprises. *International Journal of Manufacturing Technology and Management (IJMTM)*. 2016. vol. 30. no. 5. pp. 306–325. DOI: 10.1504/IJMTM.2016.078918.
2. Landau I.D., Zito G. *Digital Control Systems, Design, Identification and Implementation*. Springer, 2006. 484 p.
3. Astrom J., Wittenmark B. *Computer Controlled Systems: Theory and Design*. H.J.: Prentice Hall, 2002. 557 p.
4. Kilian C.T. *Modern control technology: Components and systems*. Novato, CA: Delmar Thomson Learning, 2000. 608 p.

5. Babishin V., Taghipour S. Optimal maintenance policy for multicomponent systems with periodic and opportunistic inspections and preventive replacements. *Applied mathematical modelling*. 2016. vol. 40. no. 23. pp. 10480–10505. DOI: 10.1016/j.apm.2016.07.019.
6. Akimenko T.A., Larkin E.V., Bogomolov A.V., Privalov A.N. [Model of formation of a scene image in the television module of a mobile robot moving over rough terrain]. *Informacionno-upravljajushhie sistemy – Information and control systems*. 2024. no. 5(132). pp. 16–24. DOI: 10.31799/8853-2024-5-16-24. (In Russ.).
7. Larkin E., Privalov A., Bogomolov A., Akimenko T. Digital control of continuous production with dry friction at actuators. *Smart Innovation, Systems and Technologies*. 2022. vol. 232. pp. 427–436.
8. Bogomolov A.V., Larkin E.V., Privalov A.N. Mathematical models of software failures of digital control systems. *Automatic Documentation and Mathematical Linguistics*. 2025. vol. 59. no. 1. pp. 1–8. DOI: 10.3103/S000510552470041.
9. Larkin E.V., Bogomolov A.V., Privalov A.N., Akimenko T.A. [Mathematical support for hierarchical digital control of a complex technological object]. *Vestnik JurGU. Serija: Matematika. Mehanika. Fizika – Bulletin of YuSU. Series: Mathematics. Mechanics. Physics*. 2024. vol. 16. no. 4. pp. 43–55. (In Russ.).
10. Harada T, Alba E. Parallel genetic algorithms: a useful survey. *ACM Computing Survey*. 2020. vol. 53. no. (4). pp. 1–39. DOI: 10.1145/3400031.
11. Hedar A., Fukushima M. Simplex coding genetic algorithm for the global optimization of nonlinear functions, in *Multi-Objective Programming and Goal Programming*. *Advances in Soft Computing*. Springer-Verlag, 2003. pp. 135–140. DOI: 10.1007/978-3-540-36510-5_17.
12. Helal M.H.S., Fan C.C., Liu D., Yuan S. Peer-to-peer based parallel genetic algorithm. *International Conference on Information, Communication and Engineering (ICICE)*. Xiamen, 2017. pp. 535–538. DOI: 10.1109/ICICE.2017.8478917.
13. Hu C., Wang X., Mandal M.K., Meng M., Li D. Efficient face and gesture recognition techniques for robot control. *Canadian Conference on Electrical and Computer Engineering*. IEEE, Montreal, 2003. pp. 1757–1762. DOI: 10.1109/CCECE.2003.1226250.
14. Jebari K. Selection methods for genetic algorithms. *International Journal of Emerging Sciences*. 2013. no. 3. pp. 333–344.
15. Hamann A., Racu R., Ernst R. Multi-dimensional robustness optimization in heterogeneous distributed embedded systems. *Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium, (RTAS'07)*. 2007. pp. 269–280. DOI: 10.1109/RTAS.2007.19.
16. Briat C. Stability and performance analysis of linear positive systems with delays using input-output methods. *International Journal of Control*. 2018. vol. 91. no. 7. pp. 1669–1692. DOI: 10.1080/00207179.2017.1326628.
17. Islam M., Chen G., Jin S. An overview of neural network. *American Journal of Neural Network and Application*. 2019. vol. 5. no. 1. pp. 7–11. DOI: 10.11648/j.ajnn.20190501.12.
18. Pavlov A.V. About the equality of the transform of Laplace to the transform of Fourier. *Issues of Analysis*. 2016. vol. 5. no. 4. pp. 21–30.
19. Li J., Farquharson C.G., Hu X. Three effective inverse Laplace transform algorithms for computing time -domain electromagnetic responses. *Geophysics*. 2015. vol. 81. no. 2. DOI: 10.1190/geo2015-0174.1.
20. Meyer-Baese U. *Digital signal processing*. Springer-Verlag Berlin, Heidelberg, 2004. 527 p.

21. Yeh Y.-C., Chu Y., Chiou C.W. Improving the sampling resolution of periodic signals by using controlled sampling interval method. *Computers & Electrical Engineering*. 2014. vol. 40. no. 4. pp. 1064–1071. DOI: 10.1016/j.compeleceng.2013.12.002.
22. Ang K.N., Chong G., Li Y. PID control system analysis, design and technology. *IEEE Transactions of control systems technology*. 2005. vol. 13. no. 4. pp. 559–576. DOI: 10.1109/TCST.2005.847331.
23. O'Dwier A. PID compensation of time delay processes 1999-2002: a survey. *Proceedings of the American control conference*. 2003. pp. 1494–1499. DOI: 10.21427/dkx2-9503.

Larkin Eugene — Ph.D., Dr.Sci., Professor, Professor-consultant of the department, Department of industrial automation and robotics, Tula State University. Research interests: stability theory of hierarchical multi-loop control systems for complex technological objects. The number of publications — 350. elarkin@mail.ru; 92, Lenin Ave., 300012, Tula, Russia; office phone: +7(487)273-4444.

Bogomolov Aleksey — Ph.D., Dr.Sci., Professor, Leading researcher, Federal Research Center «Computer Science and Control» of the Russian Academy of Sciences. Research interests: digital technologies for collecting and processing information. The number of publications — 350. a.v.bogomolov@gmail.com; 44/2, Vavilova St., 119333, Moscow, Russia; office phone: +7(499)135-6260.

Privalov Aleksandr — Ph.D., Dr.Sci., Professor, Director of the institute, Institute of advanced information technologies, Tula State Lev Tolstoy Pedagogical University. Research interests: methods of designing software for information systems, training systems, computer modeling, distributed computing. The number of publications — 350. privaloov.61@mail.ru; 125a, Lenin Ave., 300026, Tula, Russia; office phone: +7(487)235-1488.

Acknowledgements. This research is supported by Russian Science Foundation (project No. 24-29-20256).

Д.И. КРАСНОВ, М.А. ВОЛЫНСКИЙ, А.А. ГУСЕВ
**ОЦЕНКА СОСТОЯНИЯ АГЕНТА С ДИНАМИКОЙ
СТОХАСТИЧЕСКОГО ХАРАКТЕРА С ПОМОЩЬЮ
РЕКУРРЕНТНЫХ ФИЛЬТРОВ**

Краснов Д.И., Волынский М.А., Гусев А.А. Оценка состояния агента с динамикой стохастического характера с помощью рекуррентных фильтров.

Аннотация. В работе представлен метод оценки состояния агента многоагентной биологической системы по зашумленным измерениям с помощью рекуррентных фильтров. Рассмотрена проблема масштабируемости существующих подходов к регистрации поведения лабораторных грызунов и отсутствия единого математического описания. Приведено математическое описание агента биологической системы как нелинейной дискретной стохастической системы в пространстве состояний и постановка задачи оценки его состояния. Представлено решение поставленной задачи на примере ключевых точек крысы линии Wistar, которые измеряются с помощью предварительно обученного детектора. В работе предлагается использовать полносвязную нейронную сеть для параметризации неизвестной динамики системы. Для сравнительного анализа выбраны фильтр частиц (последовательный метод Монте-Карло) и ансамбльный фильтр Калмана. Сравнение методов производилось на собранном и подготовленном наборе данных, который содержит изображения с разрешением 1060×548 и разметку скелетных точек крысы. Представленные результаты экспериментов по сравнению методов оценки состояния показывают высокую эффективность предложенного метода и его преимущество перед аналитическим описанием нелинейной динамики системы. Среди сравниваемых подходов минимальную среднюю ошибку 6,4 пикселей показала двойная оценка вектора состояния и параметров нейронной сети с помощью двух ансамбльных фильтров Калмана. Однако для практического применения в реальном времени больше подходит один фильтр с предварительно обученной нейронной сетью. В таких условиях ансамбльный фильтр Калмана показал лучшие результаты по точности, чем фильтр частиц (средняя ошибка 8,1 пикселя против 12,0 пикселей). Результаты работы могут быть использованы для решения задачи автоматической регистрации поведения крыс линии Wistar с помощью параметризации функций, связывающих векторы состояния с выходными векторами индивидуального и группового поведения.

Ключевые слова: ключевые точки скелета, лабораторные животные, многоагентная биологическая система, ансамбльный фильтр Калмана, фильтр частиц, нейронная сеть, оценка вектора состояния.

1. Введение. Многоагентный подход в современном системном анализе, теории управления и искусственном интеллекте предполагает использование децентрализованных агентов для моделирования и управления сложными системами. В рамках такого подхода под агентом понимается реальная или виртуальная сущность, функционирующая во внешней среде. Агента можно описать с помощью системы, которая имеет определенный набор состояний и перечень возможных действий. При этом агент способен проявлять как индивидуальное независимое поведение, так и взаимодействовать с другими агентами, проявляя групповое или социальное поведение.

Многоагентный подход или агентное моделирование позволяет описывать достаточно широкий класс сложных систем, переходя от простейшего взаимодействия между агентами к общему моделированию всей системы. При этом существует множество различных прикладных задач, для которых применяется многоагентное моделирование [1], но в контексте данной работы удобно разделить эти задачи на две группы. К первой группе относятся задачи, связанные с моделированием и симуляцией развития и динамики сложных систем. Примером таких задач является анализ экологических систем, динамики популяции животных, симуляция биологических систем [2] или поведения пешеходов с помощью динамики социального взаимодействия [3]. Для таких задач характерно большое количество агентов, имеющих достаточно простое математическое описание, позволяющее эффективно масштабировать модель, моделируя развитие системы. В этом случае на первый план выходит глобальное состояние многоагентной системы (МАС) и ее макроскопические параметры, а агент имеет вспомогательную роль.

Другой задачей является распределенное или сетевое управление. Обзорные работы по сетевому управлению [4, 5] рассматривают МАС с позиции классической теории управления. В рамках такого представления агент характеризуется с помощью динамической системы с входом и выходом, например в пространстве состояний, а взаимодействие между агентами описывается с помощью графа, ребра которого представляют связи между отдельными агентами. Динамика агента при этом описывается простой системой, например интегратором, а задачей является синтез законов распределенного управления для роя, следования за лидером, следования в заданную точку или исследования неизвестного пространства. Распределенность и децентрализованность управления в таких задачах обусловлены двумя факторами. Во-первых, агенты являются относительно автономными сущностями, а связи между ними возникают только в момент управления, то есть граф связей изменяется во времени [5]. Во-вторых, каждый агент обладает информацией только о ближайших или соседних агентах, что является одной из сложностей сетевого управления.

Такое представление характерно для задач коллективного управления беспилотными летательными аппаратами [6 – 9] или наземными роботами [10, 11]. При таком подходе поведение отдельно взятого агента так же не является существенным, поскольку основной идеей сетевого управления является достижение коллективной цели. Чаще всего динамика агентов при таком подходе известна, так как

речь идет о программируемых роботах, однако существуют работы [12], посвященные оптимальным алгоритмам управления МАС с использованием параметрической идентификации с помощью линейной регрессионной модели.

В качестве другого примера задачи коллективного управления и класса МАС, в которых возникает такая задача, можно выделить футбол роботов. Данный пример интересен сравнительно малым количеством взаимодействующих агентов и вниманием к индивидуальному поведению каждого агента, что отличает футбол роботов от задач управления группой беспилотных летательных аппаратов. Внимание к описанию индивидуального поведения необходимо, поскольку динамика и закон управления агентами команды противника неизвестны. Для футбола роботов характерны задачи оптимального расположения агентов на поле (сетевое управление) [13, 14] и объезда движущегося препятствия (индивидуальное управление) [15]. Основной сложностью для синтеза закона управления в футболе роботов является неопределенность в движении агентов команды противника. Поэтому большинство законов индивидуального управления сводится в детектировании агента противника и объезда его как препятствия, согласно заранее определенной динамике движения.

В контексте данной работы интерес представляют системы, имеющие как сложную или неизвестную стохастическую динамику, так и сложное взаимодействие между агентами. Например, в работе [16] рассматривается система из корабля и нескольких буксиров, задача управления которой заключается в перемещении корабля в определенную точку посредством тяги буксиров под воздействием возмущений окружающей среды. Другим примером является управление роботизированной репликой рыбы с помощью обратной связи [17 – 21]. Данная задача представляет особый интерес, поскольку роботизированная реплика рыбы должна имитировать под естественные движения живой рыбы, чтобы не вызывать отторжение у живых рыб в ходе исследования взаимодействия между ними. Поэтому необходимо учитывать как индивидуальное поведение реплики, то есть траекторию движения, так и групповое поведение, то есть реакцию на движение других рыб с помощью обратной связи. Обратная связь в данной задаче заключается в адаптации движения роботизированной рыбы к траектории движения и поведению живых рыб, которые анализируются с помощью обработки видеопотока с камеры в реальном времени. Основной сложностью является моделирование неизвестной стохастической динамики живой рыбы,

что предполагает проработку сложного математического описания [22 – 25], основанного на анализе собранных экспериментальных данных, поскольку детерминированное описание в данной задаче невозможно. Анализ траекторий движений рыбы *Danio rerio* показал, что изменение ее угловой скорости во времени обладает определенной авторегрессионностью [23]. Это позволило моделировать изменение угловой и линейной скоростей с помощью стохастического процесса Орнштейна-Уленбека, параметры которого определялись с помощью метода максимального правдоподобия.

В случае роботизированной реплики рыбы задача управления сводится к моделированию траектории движения, в которой необходимо учесть периоды неподвижности, движение вдоль стенок аквариума и движение при наличии других рыб. Однако, существует класс МАС, агенты которых имеют более сложное индивидуальное поведение и групповое взаимодействие. Сложность заключается в широком перечне возможных поведенческих активностей, которые выходят за рамки траекторного описания. Для таких систем не характерна задача управления, поскольку на передний план выступает задача анализа как поведения отдельного агента, так и взаимодействия между агентами. Ярким примером являются многоагентные биологические системы, например система лабораторных грызунов, взаимодействующих в закрытом боксе. Примером задачи анализа таких систем является задача автоматизации длительного мониторинга поведения лабораторных животных с помощью компьютерного зрения [26, 27]. Отличительной особенностью в данной задаче является агентоориентированный подход, в котором поведение отдельного агента, эволюция его поведенческого профиля и межагентное взаимодействие имеют большее значение, чем динамика всей системы в целом. Такой подход позволяет не только объединить имеющиеся решения по мониторингу поведения лабораторных грызунов [28 – 30], но и обеспечить масштабируемость между различными биологическими системами, что в настоящее время затруднено из-за высокой степени специализации каждого решения.

Другой сложностью анализа таких МАС, как и в задаче управления роботизированной репликой рыбы, является неизвестная стохастическая динамика системы. Результаты работ [31, 32] показывают, что в пространственных траекториях движения лабораторных грызунов существуют взаимосвязи и паттерны, которые возможно выделить с помощью методов машинного обучения. Однако математическое описание стохастической динамики лабораторного грызуна является сложной задачей, которая может быть решена только

с помощью подходов, связанных с использованием большого количества экспериментальных данных. Например, выделенные в ходе анализа экспериментальных данных паттерны движения мыши могут быть использованы для построения стохастической модели движения животного, аналогично модели движения рыбы в работе [23]. Отдельным вопросом для исследования является переносимость и масштабируемость такой модели не только на различные виды грызунов (мыши и крысы), но даже на различных особей одного вида, поскольку каждая особь обладает своим индивидуальным поведенческим профилем, который может не соответствовать модели.

Как было отмечено ранее, сложность и многоуровневость поведения агентов также является особенностью, которую необходимо учитывать при разработке алгоритмов анализа таких МАС. В существующих подходах к анализу поведения агентов на примере лабораторных грызунов прослеживается узкая направленность не только по типам взаимодействия (индивидуальное и социальное), но и по множеству возможных действий внутри одного типа взаимодействия. Например, в работе [28] анализируется социальное взаимодействие двух мышей на примере трех классов: нападение, запрыгивание на другую особь, обнюхивание. В работе [33] представлен анализ индивидуального поведения белой крысы на примере классов: питье, прием пищи, груминг, обнюхивание, ходьба, прыжок, отдых и вставание на задние лапы. Однако среди существующих подходов не найдены решения, позволяющие одновременно регистрировать как индивидуальное поведение агента, так и социальное взаимодействие между агентами.

На основе проведенного анализа можно сделать вывод о том, что большинство существующих подходов к анализу, моделированию и управлению МАС рассматривают системы с большим количеством агентов, имеющих сравнительно простое математическое описание и известную динамику, что позволяет их эффективно масштабировать. Лишь небольшое количество работ посвящено анализу МАС, характеризующихся малым количеством агентов со сложной или неизвестной стохастической динамикой и сложным взаимодействием между агентами.

Поэтому целью данной работы является постановка и решение задачи оценки состояния агента МАС по зашумленным измерениям на примере крысы линии Wistar. Оценка состояния необходима для анализа индивидуального поведения агента и взаимодействия между агентами, как будет показано в дальнейшем. Новизна данной работы заключается в достаточно общей постановке задачи анализа МАС,

которая включает в себя описание дискретной нелинейной стохастической системы в пространстве состояний, где в качестве вектора состояния выступает вектор ключевых точек, а в качестве выходного вектора – вектор вероятности совершения действия из множества возможных. Кроме того, представленный вектор вероятности поведений может быть разделен на индивидуальное и коллективное типы поведения, переключаемые с помощью весовой функции. Описанный подход применим для анализа МАС, для которых характерен вероятностный характер динамики и сложное взаимодействие между агентами, предполагающее широкий спектр возможных действий. В случае автоматической регистрации поведения лабораторных грызунов представленный подход может объединить анализ как индивидуального поведения агента, так и социального взаимодействия между агентами, что будет способствовать повышению качества медицинских исследований.

В настоящей статье представлено продолжение предыдущего исследования по оценке вектора состояния биологической системы по зашумленным измерениям [34]. Целью предыдущей работы было определить наиболее подходящее аналитическое описание динамики агента биологической системы. В ходе исследования был получен набор данных с траекториями ключевых точек крысы линии Wistar, расширенная версия которого используется в данной работе. А результаты исследования показали, что ни одна из детерминированных аналитических моделей движения не может в полной мере описать динамику биологического агента. Поэтому в данной работе для параметризации модели динамики системы предлагается использовать полносвязную нейронную сеть. Приведенные результаты показывают преимущество описанного подхода по сравнению с аналитической нелинейной моделью движения.

2. Постановка задачи. Рассмотрим в качестве описания i -го агента нелинейную дискретную систему в пространстве состояний:

$$\begin{aligned} \mathbf{X}_i(k+1) &= \mathbf{f}(\mathbf{X}_i(k), \mathbf{w}(k)) \\ \begin{bmatrix} \mathbf{y}_i^{\text{ind}}(k) \\ \mathbf{y}_{ij}^{\text{soc}}(k) \end{bmatrix} &= \begin{bmatrix} (1 - \max(\alpha_{ij}(k))) \mathbf{h}^{\text{ind}}(\mathbf{X}_i(k)) \\ \alpha_{ij}(k) \mathbf{h}^{\text{soc}}(\mathbf{X}_i(k), \mathbf{X}_j(k)) \end{bmatrix}, \end{aligned} \quad (1)$$

где $\mathbf{X}_i(k) = [\mathbf{x}_i(k - K_{\text{prev}}) \dots \mathbf{x}_i(k)]^T$ – вектор, составленный из векторов состояния $\mathbf{x}_i(k)$ i -го агента за K_{prev} предыдущих моментов

времени, где $i=1, \dots, N$; k – номер отсчета, где $k=1, \dots, K$; $\mathbf{X}_i(0) = \mathbf{X}_{i0}$ – начальное состояние системы; $\mathbf{w}(k)$ – вектор шума, обусловленный ошибкой математической модели; $\mathbf{f}(\cdot)$ – нелинейная функция, описывающая динамику системы; $\mathbf{y}_i^{\text{ind}}(k)$ – недоступный для измерения выходной вектор индивидуального поведения i -го агента; $\mathbf{y}_{ij}^{\text{soc}}(k)$ – недоступный для измерения выходной вектор группового поведения i -го и j -го агентов, где $i, j=1, \dots, N$, $i \neq j$; $\mathbf{h}^{\text{ind}}(\cdot)$ – неизвестная нелинейная функция, связывающая предыдущие K_{prev} векторов состояния i -го агента с вектором индивидуального поведения i -го агента; $\mathbf{h}^{\text{soc}}(\cdot)$ – неизвестная нелинейная функция, связывающая предыдущие K_{prev} векторов состояния i -го и j -го агентов с вектором группового поведения i -го и j -го агентов; α_{ij} – коэффициент связи между i -м и j -м агентами; N – общее количество агентов; K – общее количество отсчетов.

По системе (1) можно увидеть, что агенты являются однородными, то есть функции $\mathbf{f}(\cdot)$, $\mathbf{h}^{\text{ind}}(\cdot)$ и $\mathbf{h}^{\text{soc}}(\cdot)$ одинаковы для каждого агента. Функции, описывающие индивидуальные и коллективные действия агента независимы. Более того, индивидуальные и коллективные действия агента в момент времени k исключают друг друга, то есть агент может совершать либо индивидуальные действия, либо коллективные, что определяется коэффициентами связи. Коэффициенты связи между агентами α_{ij} представляют собой аналог ребер графа, который используется для описания попарного взаимодействия между агентами в алгоритмах сетевого управления. В данной работе предлагается вычислять коэффициенты связи как смещенную и отраженную функцию Хевисайда, которая реализует переключение между индивидуальным и групповым поведением:

$$\alpha_{ij}(k, d_{ij}) = \begin{cases} 1, & d_{ij} < d_{\max} \\ 0, & d_{ij} \geq d_{\max} \end{cases}, \quad (2)$$

где d_{ij} – Евклидово расстояние между центрами масс i -го и j -го агентов; $d_{\max}$ – максимальное расстояние, при котором взаимодействие считается парным.

Анализ многоагентной биологической системы в данном контексте заключается в определении неизвестных выходных векторов $\mathbf{y}_i^{\text{ind}}(k)$ и $\mathbf{y}_{ij}^{\text{soc}}(k)$, представляющих собой векторы вероятностей совершения агентом того или иного действия из заранее определенных множеств индивидуальных и социальных активностей. Определить эти

векторы возможно только с помощью решения задачи идентификации неизвестных нелинейных функций $\mathbf{h}^{\text{ind}}(\cdot)$ и $\mathbf{h}^{\text{soc}}(\cdot)$ на основе предыдущих оценок векторов состояния и истинных значений выходных векторов. Решение этой задачи выходит за рамки данной работы, поскольку она посвящена задаче оценки вектора состояния по зашумленным измерениям.

Вектор состояния в задаче анализа поведения агента биологической системы может представлять собой координаты ключевых точек скелета грызуна за последние K_{prev} кадров. Рассмотрим задачу оценки вектора состояния агента. Для простоты ограничимся рассмотрением автономного агента (крыса находится в боксе одна). Тогда математическое описание системы (1) примет вид:

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{f}(\mathbf{X}(k), \mathbf{w}(k)) \\ \mathbf{z}(k) &= \mathbf{H}\mathbf{X}(k) + \mathbf{v}(k) \end{aligned} \quad (3)$$

где $\mathbf{H}$ – матрица измерения, связывающая векторы состояния с вектором измеряемых параметров $\mathbf{z}(k)$; $\mathbf{v}(k)$ – белый гауссовский шум измерений с нулевым математическим ожиданием и ковариационной матрицей $\mathbf{R}$; k – номер отсчета, где $k=1, \dots, K$; K – общее количество отсчетов.

Стоит отметить, что функция динамики системы $\mathbf{f}(\cdot)$ может быть задана несколькими способами. Например, в предыдущей работе [34] было предложено использовать линейную модель равноускоренного движения [35] и нелинейную модель Constant Turn Rate and Acceleration (CTRA) [36]. В данной работе предлагается параметризовать динамику системы с помощью полносвязной нейронной сети или многослойного перцептрона (MLP):

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{f}(\mathbf{X}(k), \boldsymbol{\theta}, \mathbf{w}(k)) \\ \mathbf{z}(k) &= \mathbf{H}\mathbf{X}(k) + \mathbf{v}(k) \end{aligned} \quad (4)$$

где $\boldsymbol{\theta}$ – вектор параметров; k – номер отсчета, где $k=1, \dots, K$; K – общее количество отсчетов.

Вектор параметров $\boldsymbol{\theta}$ содержит веса полносвязной нейронной сети, а его размерность зависит от длины вектора состояния, количества слоев сети и количества нейронов в скрытых слоях (в данной работе его размерность равна 866). Этот вектор может быть определен либо с помощью градиентных методов по критерию минимизации оценочного функционала (среднеквадратическая

ошибка) на основе собранных ранее экспериментальных данных, либо с помощью двойной оценки в процессе работы рекуррентного фильтра, как это предложено в работе [37].

3. Материалы и методы. В данной работе был использован тот же набор данных, что в работе [34]. В качестве вектора состояния агента $\mathbf{x}(k)$ использован вектор, составленный из координат x , y ключевых точек скелетной модели крысы (рисунок 1), которые были выбраны экспертом по поведению. Скелетная модель крысы состоит из 13-ти точек: нос (1); правый и левый глаз (2 и 3); правое и левое ухо (5 и 6); средняя точка головы (4); средняя точка пояса передних конечностей (7); средняя точка позвоночника (8); правая и левая передняя конечность (9 и 10); основание хвоста (11); правая и левая задняя конечность (12 и 13). Набор данных содержит около 3 тыс. изображений с разрешением 1060×548 и разметкой ключевых точек скелета. Стоит отметить, что исходные видеозаписи сняты в так называемой обогащенной среде, то есть в бокс с крысой дополнительно помещены веревки и домики из органического стекла, что создает дополнительные помехи и окклюзии отдельных частей крысы.

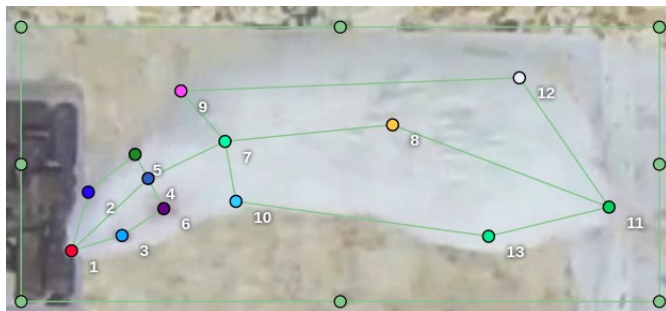


Рис. 1. Расположение точек скелета на крысе [34]

В качестве вектора $\mathbf{X}(k)$ используется вектор, составленный из ключевых точек за K_{prev} предыдущих кадров. В данной работе использовалось три последних кадра. Вектором зашумленных измерений $\mathbf{z}(k)$ являются предсказания детектора ключевых точек [38], обученного на экспериментальных данных. Детектор точек предсказывает точки в прямоугольной области кадра с крысой, которая выделяется с помощью обученного детектора объектов [39]. В качестве прямоугольника, ограничивающего зону кадра с крысой во время обучения детектора объектов, был использован прямоугольник минимального размера, содержащий все ключевые точки крысы на

данном кадре, поскольку отдельная разметка прямоугольной области с крысой не производилась. Тогда ковариационной матрицей шума измерения $\mathbf{R}$ является диагональная матрица, диагональ которой составлена из дисперсий ошибок детектора точек для координат x и y каждой точки:

$$\text{diag}(\mathbf{R}) = \begin{bmatrix} \sigma_{x1}^2 & \sigma_{y1}^2 & \dots & \sigma_{xN_{kp}}^2 & \sigma_{yN_{kp}}^2 \end{bmatrix}, \quad (5)$$

где σ_{xi}^2 и σ_{yi}^2 – дисперсии ошибок детектора точек для координат x и y соответственно; N_{kp} – количество ключевых точек (в данной работе 13).

Например, для координаты x для i -й точки:

$$\sigma_{xi}^2 = \frac{1}{M-1} \sum_{j=1}^M (\Delta x_i(j) - \Delta x_{im})^2, \quad (6)$$

где M – количество изображений в выборке; $\Delta x_i(j)$ – разность между истинным и предсказанным сверточной сетью положением i -ой ключевой точки по координате x на j -ом изображении из выборки; Δx_{im} – среднее значение $\Delta x_i(j)$ для всех изображений из выборки.

Тогда систему (4) можно дополнить, определив матрицу $\mathbf{H}$ и приняв допущение, что шум $\mathbf{w}(k)$ так же является белым Гауссовским шумом с нулевым математическим ожиданием и ковариационной матрицей $\mathbf{Q}$:

$$\begin{aligned} [\mathbf{x}(k+1) \quad \dots \quad \mathbf{x}(k-K+1)]^T &= \mathbf{f}([\mathbf{x}(k) \quad \dots \quad \mathbf{x}(k-K)]^T, \boldsymbol{\theta}) + \mathbf{w}(k) \\ \mathbf{z}(k) &= \begin{bmatrix} \mathbf{I}_{2N_{kp} \times 2N_{kp}} & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{x}(k) \quad \dots \quad \mathbf{x}(k-K)]^T + \mathbf{v}(k) \end{aligned} \quad (7)$$

Главной задачей для оценки состояния системы является определение функции $\mathbf{f}(\cdot)$. Функция $\mathbf{f}(\cdot)$ в данной работе параметризована с помощью полносвязной нейронной сети с одним скрытым слоем, содержащим 8 нейронов. В качестве ковариационной матрицы шума модели $\mathbf{Q}$ можно использовать диагональную матрицу, элементы диагонали которой рассчитаны по выражениям (5) и (6) аналогично матрице $\mathbf{R}$ и дополнены нулями после $2N_{kp}$ -го элемента, поскольку ошибка параметризации для $\mathbf{x}(k)$ не влияет на векторы состояния за K_{prev} предыдущих кадров. В данной работе оценка

вектора состояния системы (7) проведена с помощью ансцентного фильтра Калмана (UKF) [40] и фильтра частиц (PF) или последовательного метода Монте-Карло тремя способами: UKF с определенным в процессе обучения нейронной сети вектором параметров, двойным UKF (Dual UKF) для параллельной оценки вектора параметров [37] и мультиоблачным последовательным методом Монте-Карло [41]. Для сравнения с описанными выше методами в качестве динамики системы была использована нелинейная модель CTRA с UKF, как в работе [34].

В случае использования аналитической нелинейной модели CTRA функция $\mathbf{f}(\cdot)$ не нуждается в параметризации, а вектор состояния дополняется скоростью, ускорением, углом поворота и угловой скоростью для каждой ключевой точки скелетной модели. Кроме того, не учитывается история предыдущих K_{prev} векторов состояния для обновления состояния агента, что приводит к упрощению модели.

В случае предварительного обучения полносвязной нейронной сети для параметризации динамики (методы MLP с UKF и MLP с PF) используется ориентированный на данных подход. Нейронная сеть была обучена предсказывать следующий вектор состояния (множество ключевых скелетных точек) по K_{prev} предыдущим ключевым точкам этой траектории, минимизируя среднеквадратическую ошибку между координатами предсказанной точки и истинной точки. Обученная сеть с замороженными весами использовалась в процессе работы фильтров. Для UKF использовалась реализация, предложенная в работе [40]. В качестве PF использовалась реализация последовательного мультиоблачного метода Монте-Карло, предложенная в работе [41], основная идея которого состоит в генерации нескольких облаков частиц вокруг оцененных векторов состояния с максимальным весом, вместо использования одного постоянного облака частиц. Для пересчета веса частиц использовалось распределение вероятности $\mathbf{p}(\mathbf{z}(k)|\mathbf{X}(k))$, связанное с невязкой предсказания (с учетом дальнейшего масштабирования веса каждой частицы):

$$\mathbf{p}(\mathbf{z}(k), \mathbf{X}(k)) = \exp \left(-\frac{1}{2} (\mathbf{z}(k) - \mathbf{H}\mathbf{X}(k))^T \mathbf{R}^{-1} (\mathbf{z}(k) - \mathbf{H}\mathbf{X}(k)) \right). \quad (8)$$

На каждом шаге алгоритма происходил выбор нескольких частиц с наибольшим весом, удаление остальных частиц и генерация новых частиц вокруг выбранных, для того чтобы общее количество частиц осталось неизменным.

Метод MLP с dual UKF предполагает онлайн оценку параметров параметризованной модели динамики с помощью второго UKF [37]. Таким образом, два фильтра работают совместно, оценка параметров модели от одного фильтра используется для оценки вектора состояния в другом фильтре. В таком случае систему (7) можно расширить с помощью модели параметров в пространстве состояний:

$$\begin{aligned}\boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \mathbf{I}(k) \\ \mathbf{z}(k) &= \begin{bmatrix} \mathbf{I}_{2N_{kp} \times 2N_{kp}} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{f}\left(\begin{bmatrix} \mathbf{x}(k) & \dots & \mathbf{x}(k-K) \end{bmatrix}^T, \boldsymbol{\theta}(k)\right) + \mathbf{w}(k) + \mathbf{v}(k),\end{aligned}\quad (9)$$

где $\boldsymbol{\theta}(k)$ и $\boldsymbol{\theta}(k+1)$ – векторы параметров модели в моменты времени k и $k+1$; $\mathbf{I}(k)$ – вектор белого Гауссовского шума с нулевым математическим ожиданием и ковариационной матрицей $\mathbf{L}$.

Система (9) используется одним из двух UKF для онлайн оценки параметров нейронной сети. Вектор шума $\mathbf{I}(k)$ помогает фильтру оценивать оптимальные параметры модели и введен в систему для лучшей сходимости алгоритма. Ковариационная матрица $\mathbf{L}$ представляет собой диагональную матрицу с очень маленькими числами на главной диагонали (в эксперименте порядка 10^{-6}).

Для оценки точности сравниваемых методов из экспериментального набора данных было выбрано 26 траекторий ключевых точек по 100 кадров в каждой. В качестве ошибки использовалось Евклидово расстояние между истинной точкой траектории и точкой, полученной с помощью каждого рекуррентного фильтра, усредненное между всеми точками данного кадра. Средняя ошибка определяется по выражению:

$$e_m(k) = \frac{1}{N_{kp} T} \sum_{t=1}^T \sum_{n=1}^{N_{kp}} \sqrt{(x(t, n, k) - x^*(t, n, k))^2 + (y(t, n, k) - y^*(t, n, k))^2}, \quad (10)$$

где k – номер кадра в траектории; t – номер траектории; n – индекс ключевой точки; T – общее количество траекторий; $x(t, n, k)$ и $y(t, n, k)$ – координаты x и y точки с индексом n , предсказанной на кадре k траектории t ; $x^*(t, n, k)$ и $y^*(t, n, k)$ – истинные координаты x и y точки с индексом n на кадре k траектории t .

Визуализация вычислительного эксперимента представлена на рисунке 2. Эксперимент проводился на процессоре Intel Core i5-10300H. Математические операции и рекуррентные фильтры

реализованы на языке Python с помощью библиотек PyTorch и NumPy. Обучение и запуск детектора объектов производилось с помощью библиотеки MMDetection, а детектора ключевых точек с помощью библиотеки MMPose.

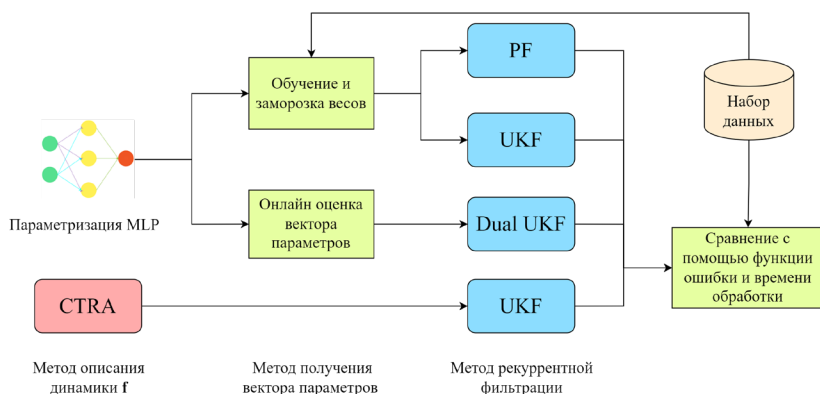


Рис. 2. Схема вычислительного эксперимента

4. Результаты и обсуждение. Результаты оценки точности каждого метода представлены как графики ошибки на каждом кадре по выражению (10). Ошибки представлены для четырех методов: UKF с параметризованной динамикой системы (MLP с UKF), двойной UKF для параллельной оценки вектора состояния и вектора параметров (MLP с dual UKF), аналитическая нелинейная модель CTRA с UKF (CTRA с UKF) и фильтр частиц с параметризованной динамикой системы (MLP с PF). На рисунке 3 представлены графики среднего значения ошибки с доверительным интервалом для каждого метода. Можно заметить характерную форму графика ошибки для MLP с dual UKF. Такая форма объясняется тем, что фильтру необходимо время для сходимости, поскольку оценка параметров модели выполняется в реальном времени. Ограниченность графиков ошибок демонстрирует сходимость описанных методов.

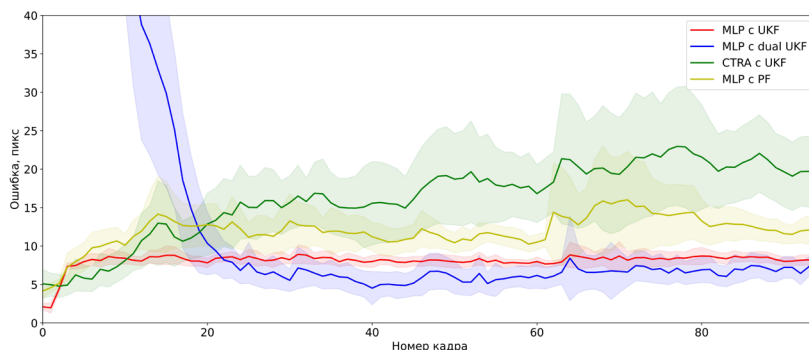


Рис. 3. Графики ошибок каждого метода

На рисунке 4 представлены коробчатые диаграммы ошибки каждого метода после точки сходимости. В качестве критерия сходимости использовалась ограниченность производной ошибки (10) (производная ошибки меньше 1,5 пикселя по модулю в течение 10-ти кадров), показывающая, что ошибка достигла своего рабочего значения и значительно не возрастает.

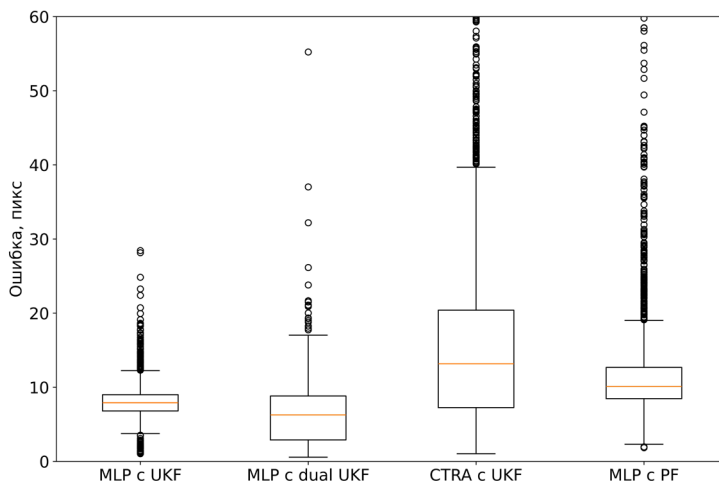


Рис. 4. Коробчатые диаграммы ошибок каждого метода

В первую очередь, результаты показывают применимость каждого из представленных методов в задаче оценки вектора состояния агента биологической системы. При этом заметно, что

параметризация динамики с помощью полносвязной нейронной сети позволяет более точно оценивать вектор состояния по сравнению с нелинейной моделью СТРА. Кроме того, такой метод в комбинации с UKF или фильтром частиц требует меньше времени на обработку, чем СТРА с UKF. В таблице 1 представлено время обработки одного кадра для каждого метода, математическое ожидание, среднее квадратическое отклонение (СКО) ошибки после сходимости и время сходимости. Результаты в таблице для PF приведены для 10 тыс. частиц. Ограниченность ошибки и небольшие значения СКО показывают сходимость и устойчивость рекуррентных фильтров, примененных в данной работе.

Таблица 1. Время обработки одного кадра и ошибка

Метод	MLP с UKF	MLP с Dual UKF	СТРА с UKF	MLP с PF
Время, мс	12	142	50	57
Мат. ожидание ошибки, пикс	8,1	6,4	16,3	12,0
СКО ошибки, пикс	2,3	4,4	13,7	7,9
Время сходимости (номер отсчета)	13	28	11	13

Таблица 1 показывает, что оценка состояния с помощью UKF с предобученной нейронной сетью для параметризации динамики имеет меньшую ошибку (8,1 пикс) и меньшую скорость обработки одного кадра (12 мс), чем PF (12,0 пикс и 57 мс). Отдельного внимания заслуживает метод с параллельной оценкой параметров нейронной сети с помощью двойного UKF. Этот метод показал наименьшую ошибку на тестовых данных (6,4 пикс) и наибольшее время обработки (142 мс), что делает его применение для обработки в реальном времени затруднительным. Наибольшую ошибку оценки состояния показал метод, основанный на аналитической нелинейной модели СТРА (16,3 пикс), что показывает преимущество параметризации динамики для МАС такого класса.

Однако в задачах анализа поведения лабораторных животных чаще всего необходима обработка видеозаписей, что не накладывает строгих ограничений на время обработки одного кадра и время выдачи результата, поэтому для данной задачи справедливо считать двойной UKF для оценки вектора состояния и вектора параметров модели лучшим. В случае необходимости обработки в реальном

времени возможно использовать одиночный UKF, высокая скорость обработки которого позволяет производить вычисления в том числе на одноплатных микрокомпьютерах, таких как NVIDIA Jetson Nano или RK3588.

5. Заключение. В данной работе представлено математическое описание агента в МАС со сложной стохастической динамикой и разделением индивидуального и группового поведения. Представленное описание предполагает агентоориентированный подход, для которого индивидуальное поведение агента и взаимодействие между агентами важнее, чем глобальная эволюция системы. Приведена постановка задачи оценки вектора состояния агента по зашумленным измерениям и ее решение на примере ключевых точек крысы линии Wistar. Показана применимость рекуррентных фильтров с параметризацией динамики системы с помощью полносвязной нейронной сети для решения поставленной задачи. Показано, что параметризация динамики системы позволяет уменьшить ошибку оценки вектора состояния по сравнению с использованием аналитической нелинейной модели СТРА.

Среди сравниваемых методов наиболее точной оказалась двойная оценка состояния и параметров модели с помощью двух UKF (ошибка 6,4 пикс), однако такой подход является самым ресурсозатратным и может быть применен в случае отсутствия строгих требований к скорости обработки одного кадра и скорости выдачи результата. Использование обученной нейронной сети с неизменяемыми параметрами с UKF позволило произвести оценку состояния с меньшей ошибкой, чем при использовании PF (ошибки 8,1 и 12,0 пикс соответственно), а короткое время обработки одного кадра (12 мс) позволяет использовать метод для обработки в реальном времени.

Главным направлением будущей работы является идентификация функций $\mathbf{h}^{\text{ind}}(\cdot)$ и $\mathbf{h}^{\text{soc}}(\cdot)$ на основе собранных данных для установления связи между оценками вектора состояния за предыдущее время и выходным вектором. Это позволит решить общую задачу анализа МАС представленного типа, поскольку выходные векторы представляют собой вероятности совершения различных видов индивидуального и группового поведения.

Использованный в работе набор данных, содержащий изображения и размеченные ключевые точки скелета крысы линии Wistar, является уникальным по отношению к имеющимся аналогам для мышей и крыс другого цвета. Набор данных вместе с исходным

кодом может быть предоставлен для исследовательских целей по запросу (Краснов Д.И., dmitriy_krasnov@outlook.com).

Авторы выражают благодарность д-ру мед. наук Алексею Андреевичу Шмонину (Первый Санкт-Петербургский государственный медицинский университет им. акад. И.П. Павлова) за постановку задачи, предоставление записей видеонаблюдения за поведением крыс и экспертное мнение при разметке ключевых точек скелета.

Литература

1. Кузнецов А.В. Краткий обзор многоагентных моделей // Управление большими системами: сборник трудов. 2018. № 71(5). С. 6–44. DOI: 10.25728/ubs.2018.71.1.
2. Bernardi S., Scianna M. An agent-based approach for modelling collective dynamics in animal groups distinguishing individual speed and orientation // Philosophical Transactions of the Royal Society B. 2020. vol. 375. DOI: 10.1098/rstb.2019.0383.
3. Rashid M.M., Seyedi M., Jung S. Simulation of pedestrian interaction with autonomous vehicles via social force model // Simulation Modelling Practice and Theory. 2024. vol. 132. DOI: 10.1016/j.simpat.2024.102901.
4. Antonelli G. Interconnected dynamic systems: An overview on distributed control // IEEE Control Systems Magazine. 2013. vol. 33. no. 1. pp. 76–88. DOI: 10.1109/MCS.2012.2225929.
5. Proskurnikov A.V., Fradkov A.L. Problems and methods of network control // Autom Remote Control. 2016. vol. 77. pp. 1711–1740. DOI: 10.1134/S0005117916100015.
6. Xue Y., Chen W. Multi-Agent Deep Reinforcement Learning for UAVs Navigation in Unknown Complex Environment // IEEE Transactions on Intelligent Vehicles. 2024. vol. 9. no. 1. pp. 2290–2303. DOI: 10.1109/TIV.2023.3298292.
7. Zhao X., Yang R., Zhong L., Hou Z. Multi-UAV Path Planning and Following Based on Multi-Agent Reinforcement Learning // Drones. 2024. vol. 8. no. 1. DOI: 10.3390/drones8010018.
8. Shi H., Zhao Z., Chen J., Zhou M., Liu Y. Enhancing Unmanned Aerial Vehicle Path Planning in Multi-Agent Reinforcement Learning through Adaptive Dimensionality Reduction // Drones. 2024. vol. 8. no. 10. DOI: 10.3390/drones8100521.
9. Podelenyuk, P., Epifantsev I. Modeling the Formation of a Multi-Agent System as a System of Points of a Rigid // IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). 2021. pp. 594–597. DOI: 10.1109/ElConRus51938.2021.9396179.
10. Cong Y., Feng Z., Song H., Wang S. Containment control of singular heterogeneous multi-agent systems // Journal of the Franklin Institute. 2018. vol. 355. no. 11. pp. 4629–4643. DOI: 10.1016/j.jfranklin.2018.04.009.
11. Thummalapeta M., Liu Y.-C. Survey of containment control in multi-agent systems: concepts, communication, dynamics, and controller design // International Journal of Systems Science. 2023. vol. 54. no. 14. pp. 2809–2835.
12. Luo R., Peng Z., Hu J. On Model Identification Based Optimal Control and its Applications to Multi-Agent Learning and Control // Mathematics. 2023. vol. 11. no. 4. DOI: 10.3390/math11040906.
13. Iarosh D., Reneva G., Kornilova A., Kononov P. Multiagent System of Mobile Robots for Robotic Football // 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS). 2019. pp. 1–3. DOI: 10.23919/ICINS.2019.8769365.

14. Ribeiro A.F.A., Lopes A.C.C., Ribeiro T.A., Pereira N.S.S.M., Lopes G.T., Ribeiro A.F.M. Probability-Based Strategy for a Football Multi-Agent Autonomous Robot System // *Robotics*. 2024. vol. 13. no. 1. DOI: 10.3390/robotics13010005.
15. Dewantara B.S.B., Ariyadi B.N.D. Adaptive Behavior Control for Robot Soccer Navigation Using Fuzzy-based Social Force Model // *Smart Science*. 2021. vol. 9. no. 1. pp. 14–29. DOI: 10.1080/23080477.2021.1871799.
16. Du Z., Negenborn R.R., Reppa V. Cooperative Multi-Agent Control for Autonomous Ship Towing Under Environmental Disturbances // *IEEE/CAA Journal of Automatica Sinica*. 2021. vol. 8. no. 8. pp. 1365–1379. DOI: 10.1109/JAS.2021.1004078.
17. Bonnet F., Gribovskiy A., Halloy J., Mondada F. Closed-loop interactions between a shoal of zebrafish and a group of robotic fish in a circular corridor // *Swarm Intelligence*. 2018. vol. 12. no. 3. pp. 227–244. DOI: 10.1007/s11721-017-0153-6.
18. Kim C., Ruberto T., Phamduy P., Porfiri M. Closed-loop control of zebrafish behaviour in three dimensions using a robotic stimulus // *Scientific Reports*. 2018. vol. 8. no. 1. DOI: 10.1038/s41598-017-19083-2.
19. Kopman V., Laut J., Polverino G., Porfiri M. Closed-loop control of zebrafish response using a bioinspired robotic-fish in a preference test // *Journal of The Royal Society Interface*. 2013. vol. 10. no. 78. DOI: 10.1098/rsif.2012.0540.
20. Swain D.T., Couzin I.D., Ehrlich Leonard N. Real-Time Feedback-Controlled Robotic Fish for Behavioral Experiments with Fish Schools // *Proceedings of the IEEE*. 2012. vol. 100. no. 1. pp. 150–163. DOI: 10.1109/JPROC.2011.2165449.
21. Landgraf T., Bierbach D., Nguyen H., Muggelberg N., Romanczuk P., Krause J. RoboFish: increased acceptance of interactive robotic fish with realistic eyes and natural motion patterns by live Trinidadian guppies // *Bioinspiration & Biomimetics*. 2016. vol. 11. no. 1. DOI: 10.1088/1748-3190/11/1/015001.
22. Collignon B., Séguret A., Halloy J. A stochastic vision-based model inspired by zebrafish collective behaviour in heterogeneous environments // *Royal Society Open Science*. 2016. vol. 3. no. 1. DOI: 10.1098/rsos.150473.
23. Mwafo V., Anderson R.P., Butail S., Porfiri M. A jump persistent turning walker to model zebrafish locomotion // *Journal of The Royal Society Interface*. 2015. vol. 12. no. 102. DOI: 10.1098/rsif.2014.0884.
24. Zienkiewicz A.K., Ladu F., Barton D.A.W. Data-driven modelling of social forces and collective behaviour in zebrafish // *Journal of Theoretical Biology*. 2018. vol. 443. pp. 39–51. DOI: 10.1016/j.jtbi.2018.01.011.
25. Tuqan M., Porfiri M. Mathematical Modeling of Zebrafish Social Behavior in Response to Acute Caffeine Administration // *Frontiers in Applied Mathematics and Statistics*. 2021. vol. 7. DOI: 10.3389/fams.2021.751351.
26. van Dam E.A., Noldus L.P.J.J., van Gerven M.A.J. Deep learning improves automated rodent behavior recognition within a specific experimental setup // *Journal of Neuroscience Methods*. 2020. vol. 332. DOI: 10.1016/j.jneumeth.2019.108536.
27. Grieco F., Bernstein B.J., Biemans B. et al. Measuring Behavior in the Home Cage: Study Design, Applications, Challenges, and Perspectives // *Frontiers in Behavioral Neuroscience*. 2021. vol. 15. DOI: 10.3389/fnbeh.2021.735387.
28. Segalin C., Williams J., Karigo T. et al. The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice // *eLife*. 2021. vol. 10. DOI: 10.7554/eLife.63720.
29. Hsu A.I., Yttri E.A. B-SoiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors // *Nature Communications*. 2021. vol. 12. DOI: 10.1038/s41467-021-25420-x.
30. Ru Z., Duan F. Hierarchical Spatial-Temporal Window Transformer for Pose-Based Rodent Behavior Recognition // *IEEE Transactions on Instrumentation and Measurement*. 2024. vol. 73. pp. 1–14. DOI: 10.1109/TIM.2024.3379081.

31. Weinreb C., Pearl J.E., Lin S. et al. Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics // *Nature Methods*. 2024. vol. 21. pp. 1329–1339. DOI: 10.1038/s41592-024-02318-2.
32. Donnarumma F., Prevete R., Maisto D. et al. A framework to identify structured behavioral patterns within rodent spatial trajectories // *Scientific Reports*. 2021. vol. 11. DOI: 10.1038/s41598-020-79744-7.
33. van Dam E.A., Noldus L.P.J.J., van Gerven M.A.J. Disentangling rodent behaviors to improve automated behavior recognition // *Front. Neurosci*. 2023. vol. 17. DOI: 10.3389/fnins.2023.1198209.
34. Краснов Д.И., Волинский М.А., Гусев А.А. Предобработка траекторий ключевых точек скелета в задаче автоматизации регистрации поведения лабораторных животных // *Научно-технический вестник информационных технологий, механики и оптики*. 2025. Т. 25. № 2. С. 295–302.
35. Buizza C., Fischer T., Demiris Y. Real-Time Multi-Person Pose Tracking using Data Assimilation // *IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020. pp. 438–447. DOI: 10.1109/WACV45572.2020.9093442.
36. Svensson D. Derivation of the discrete-time constant turn rate and acceleration motion model // *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. 2019. pp. 1–5. DOI: 10.1109/SDF.2019.8916654.
37. Yuan H., Han Y., Zhou Y. et al. State of Charge Dual Estimation of a Li-ion Battery Based on Variable Forgetting Factor Recursive Least Square and Multi-Innovation Unscented Kalman Filter Algorithm // *Energies*. 2022. vol. 15. no. 4. DOI: 10.3390/en15041529.
38. Huang J., Zhu Z., Guo F., Huang G. The Devil Is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation // *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020. pp. 5699–5708. DOI: 10.1109/CVPR42600.2020.00574.
39. Lyu C., Zhang W., Huang H. et al. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. 2024. arXiv preprint arXiv:2212.07784.
40. Wan E. A., Van Der Merwe R. The unscented Kalman filter for nonlinear estimation // *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. 2000. pp. 153–158. DOI: 10.1109/ASSPCC.2000.882463.
41. Волинский М.А., Гуров И.П., Скаков П.С. Рекуррентный алгоритм обработки интерферометрических сигналов на основе мультиоблачной модели предсказания // *Научно-технический вестник информационных технологий, механики и оптики*. 2014. Т. 14. № 4. С. 18–22.

Краснов Дмитрий Игоревич — аспирант, научно-образовательная лаборатория «Техническое зрение», Университет ИТМО. Область научных интересов: компьютерное зрение, распознавание образов, обработка изображений, искусственный интеллект. Число научных публикаций — 7. dmitriy_krasnov@outlook.com; Кронверкский проспект, 49, 197101, Санкт-Петербург, Россия; р.т.: +7(996)189-3418.

Волинский Максим Александрович — канд. техн. наук, доцент, директор, научно-образовательная лаборатория «Техническое зрение», Университет ИТМО. Область научных интересов: вычислительная фотоника, биофотоника, взаимодействие оптического излучения с веществом, стохастическая обработка квазипериодических сигналов. Число научных публикаций — 130. maxim.volynsky@gmail.com; Кронверкский проспект, 49, 197101, Санкт-Петербург, Россия; р.т.: +7(921)188-7946.

Гусев Александр Алексеевич — канд. техн. наук, ведущий инженер, научно-образовательная лаборатория «Техническое зрение», Университет ИТМО. Область научных интересов: машинное обучение, распознавание образов, статистическое обучение, распознавание статистических паттернов. Число научных публикаций — 32. gusew@internet.ru; Кронверкский проспект, 49, 197101, Санкт-Петербург, Россия; р.т.: +7(902)406-3854.

D. KRASNOV, M. VOLYNSKY, A. GUSEV
**STATE ESTIMATION OF AN AGENT WITH STOCHASTIC
DYNAMICS USING RECURRENT FILTERS**

Krasnov D., Volynsky M., Gusev A. State Estimation of an Agent with Stochastic Dynamics Using Recurrent Filters.

Abstract. The paper proposes a method for estimating the state vector of an agent in a multi-agent biological system based on noisy measurements using recurrent filters. It addresses the issues of scalability in existing approaches to monitoring the behavior of laboratory rodents and the absence of a unified mathematical framework. A mathematical description of an agent in the biological system is provided, along with the formulation of the task of estimating its state. The mathematical model is built upon a non-linear discrete-time system in state space. The solution to this problem is demonstrated using the example of skeletal keypoints in a Wistar rat, which are detected using a pre-trained detector. A fully connected neural network is proposed to parameterize the unknown dynamics of the system. The particle filter (a sequential Monte Carlo method) and the unscented Kalman filter were selected for a comparative analysis. The comparison of the methods was conducted on a collected and preprocessed dataset comprising images with a resolution of 1060×548 pixels and annotations of rat skeletal keypoints. The experimental results demonstrate the high efficacy of the proposed method and its advantage over an analytical description of the system's nonlinear dynamics. Among the compared methods, the dual estimation of both the state vector and the neural network parameters using two unscented Kalman filters achieved the minimal mean error of 6.4 pixels. However, for practical applications in real-time scenarios, a single filter employing a pre-trained neural network proves to be more advantageous. Moreover, the unscented Kalman filter in this case demonstrated higher accuracy than the particle filter (mean error of 8.1 pixels vs. 12.0 pixels). The results of this study can be used to solve the task of automated registration of Wistar rat behavior by parameterizing the functions that link state vectors with the output vectors of individual and group behavior.

Keywords: skeletal keypoints, laboratory animals, multi-agent biological system, unscented Kalman filter, particle filter, neural network, state vector estimation.

References

1. Kuznetsov A. [The short review of multi-agent models]. *Upravlenie bol'shimi sistemami: sbornik trudov [Large-Scale Systems Control]*. 2018. no. 71(5). pp. 6–44. DOI: 10.25728/ubs.2018.71.1. (In Russ.).
2. Bernardi S., Scianna M. An agent-based approach for modelling collective dynamics in animal groups distinguishing individual speed and orientation. *Philosophical Transactions of the Royal Society B*. 2020. vol. 375. DOI: 10.1098/rstb.2019.0383.
3. Rashid M.M., Seyed M., Jung S. Simulation of pedestrian interaction with autonomous vehicles via social force model. *Simulation Modelling Practice and Theory*. 2024. vol. 132. DOI: 10.1016/j.simpat.2024.102901.
4. Antonelli G. Interconnected dynamic systems: An overview on distributed control. *IEEE Control Systems Magazine*. 2013. vol. 33. no. 1. pp. 76–88. DOI: 10.1109/MCS.2012.2225929.
5. Proskurnikov A.V., Fradkov A.L. Problems and methods of network control. *Autom Remote Control*. 2016. vol. 77. pp. 1711–1740. DOI: 10.1134/S0005117916100015.

6. Xue Y., Chen W. Multi-Agent Deep Reinforcement Learning for UAVs Navigation in Unknown Complex Environment. *IEEE Transactions on Intelligent Vehicles*. 2024. vol. 9. no. 1. pp. 2290–2303. DOI: 10.1109/TIV.2023.3298292.
7. Zhao X., Yang R., Zhong L., Hou Z. Multi-UAV Path Planning and Following Based on Multi-Agent Reinforcement Learning. *Drones*. 2024. vol. 8. no. 1. DOI: 10.3390/drones8010018.
8. Shi H., Zhao Z., Chen J., Zhou M., Liu Y. Enhancing Unmanned Aerial Vehicle Path Planning in Multi-Agent Reinforcement Learning through Adaptive Dimensionality Reduction. *Drones*. 2024. vol. 8. no. 10. DOI: 10.3390/drones8100521.
9. Podelenyuk, P., Epifantsev I. Modeling the Formation of a Multi-Agent System as a System of Points of a Rigid. *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. 2021. pp. 594–597. DOI: 10.1109/ElConRus51938.2021.9396179.
10. Cong Y., Feng Z., Song H., Wang S. Containment control of singular heterogeneous multi-agent systems. *Journal of the Franklin Institute*. 2018. vol. 355. no. 11. pp. 4629–4643. DOI: 10.1016/j.jfranklin.2018.04.009.
11. Thummalapeta M., Liu Y.-C. Survey of containment control in multi-agent systems: concepts, communication, dynamics, and controller design. *International Journal of Systems Science*. 2023. vol. 54. no. 14. pp. 2809–2835.
12. Luo R., Peng Z., Hu J. On Model Identification Based Optimal Control and it's Applications to Multi-Agent Learning and Control. *Mathematics*. 2023. vol. 11. no. 4. DOI: 10.3390/math11040906.
13. Iarosh D., Reneva G., Kornilova A., Konovalov P. Multiagent System of Mobile Robots for Robotic Football. *26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*. 2019. pp. 1–3. DOI: 10.23919/ICINS.2019.8769365.
14. Ribeiro A.F.A., Lopes A.C.C., Ribeiro T.A., Pereira N.S.S.M., Lopes G.T., Ribeiro A.F.M. Probability-Based Strategy for a Football Multi-Agent Autonomous Robot System. *Robotics*. 2024. vol. 13. no. 1. DOI: 10.3390/robotics13010005.
15. Dewantara B.S.B., Ariyadi B.N.D. Adaptive Behavior Control for Robot Soccer Navigation Using Fuzzy-based Social Force Model. *Smart Science*. 2021. vol. 9. no. 1. pp. 14–29. DOI: 10.1080/23080477.2021.1871799.
16. Du Z., Negenborn R.R., Reppa V. Cooperative Multi-Agent Control for Autonomous Ship Towing Under Environmental Disturbances. *IEEE/CAA Journal of Automatica Sinica*. 2021. vol. 8. no. 8. pp. 1365–1379. DOI: 10.1109/JAS.2021.1004078.
17. Bonnet F., Gribovskiy A., Halloy J., Mondada F. Closed-loop interactions between a shoal of zebrafish and a group of robotic fish in a circular corridor. *Swarm Intelligence*. 2018. vol. 12. no. 3. pp. 227–244. DOI: 10.1007/s11721-017-0153-6.
18. Kim C., Ruberto T., Phamduy P., Porfiri M. Closed-loop control of zebrafish behaviour in three dimensions using a robotic stimulus. *Scientific Reports*. 2018. vol. 8. no. 1. DOI: 10.1038/s41598-017-19083-2.
19. Kopman V., Laut J., Polverino G., Porfiri M. Closed-loop control of zebrafish response using a bioinspired robotic-fish in a preference test. *Journal of The Royal Society Interface*. 2013. vol. 10. no. 78. DOI: 10.1098/rsif.2012.0540.
20. Swain D.T., Couzin I.D., Ehrlich Leonard N. Real-Time Feedback-Controlled Robotic Fish for Behavioral Experiments with Fish Schools. *Proceedings of the IEEE*. 2012. vol. 100. no. 1. pp. 150–163. DOI: 10.1109/JPROC.2011.2165449.
21. Landgraf T., Bierbach D., Nguyen H., Muggelberg N., Romanczuk P., Krause J. RoboFish: increased acceptance of interactive robotic fish with realistic eyes and natural motion patterns by live Trinidadian guppies. *Bioinspiration & Biomimetics*. 2016. vol. 11. no. 1. DOI: 10.1088/1748-3190/11/1/015001.

22. Collignon B., Séguret A., Halloy J. A stochastic vision-based model inspired by zebrafish collective behaviour in heterogeneous environments. *Royal Society Open Science*. 2016. vol. 3. no. 1. DOI: 10.1098/rsos.150473.
23. Mwaffo V., Anderson R.P., Butail S., Porfiri M. A jump persistent turning walker to model zebrafish locomotion. *Journal of The Royal Society Interface*. 2015. vol. 12. no. 102. DOI: 10.1098/rsif.2014.0884.
24. Zienkiewicz A.K., Ladu F., Barton D.A.W. Data-driven modelling of social forces and collective behaviour in zebrafish. *Journal of Theoretical Biology*. 2018. vol. 443. pp. 39–51. DOI: 10.1016/j.jtbi.2018.01.011.
25. Tuqan M., Porfiri M. Mathematical Modeling of Zebrafish Social Behavior in Response to Acute Caffeine Administration. *Frontiers in Applied Mathematics and Statistics*. 2021. vol. 7. DOI: 10.3389/fams.2021.751351.
26. van Dam E.A., Noldus L.P.J.J., van Gerven M.A.J. Deep learning improves automated rodent behavior recognition within a specific experimental setup. *Journal of Neuroscience Methods*. 2020. vol. 332. DOI: 10.1016/j.jneumeth.2019.108536.
27. Grieco F., Bernstein B.J., Biemans B. et al. Measuring Behavior in the Home Cage: Study Design, Applications, Challenges, and Perspectives. *Frontiers in Behavioral Neuroscience*. 2021. vol. 15. DOI: 10.3389/fnbeh.2021.735387.
28. Segalin C., Williams J., Karigo T. et al. The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice. *eLife*. 2021. vol. 10. DOI: 10.7554/eLife.63720.
29. Hsu A.I., Yttri E.A. B-SOId, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nature Communications*. 2021. vol. 12. DOI: 10.1038/s41467-021-25420-x.
30. Ru Z., Duan F. Hierarchical Spatial-Temporal Window Transformer for Pose-Based Rodent Behavior Recognition. *IEEE Transactions on Instrumentation and Measurement*. 2024. vol. 73. pp. 1–14. DOI: 10.1109/TIM.2024.3379081.
31. Weinreb C., Pearl J.E., Lin S. et al. Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics. *Nature Methods*. 2024. vol. 21. pp. 1329–1339. DOI: 10.1038/s41592-024-02318-2.
32. Donnarumma F., Prevete R., Maisto D. et al. A framework to identify structured behavioral patterns within rodent spatial trajectories. *Scientific Reports*. 2021. vol. 11. DOI: 10.1038/s41598-020-79744-7.
33. van Dam E.A., Noldus L.P.J.J., van Gerven M.A.J. Disentangling rodent behaviors to improve automated behavior recognition. *Front. Neurosci.* 2023. vol. 17. DOI: 10.3389/fnins.2023.1198209.
34. Krasnov D.I., Volynsky M.A., Gusev A.A. [Preprocessing of skeletal keypoints trajectories in the task of laboratory animal behavior recording automation]. *Nauchno-tehnicheskij vestnik informacionnyh tehnologij, mehaniki i optiki – Scientific and Technical Journal of Information Technologies, Mechanics and Optics*. 2025. vol. 25. no. 2. pp. 295–302. (In Russ.).
35. Buizza C., Fischer T., Demiris Y. Real-Time Multi-Person Pose Tracking using Data Assimilation. *IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020. pp. 438–447. DOI: 10.1109/WACV45572.2020.9093442.
36. Svensson D. Derivation of the discrete-time constant turn rate and acceleration motion model. *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. 2019. pp. 1–5. DOI: 10.1109/SDF.2019.8916654.
37. Yuan H., Han Y., Zhou Y. et al. State of Charge Dual Estimation of a Li-ion Battery Based on Variable Forgetting Factor Recursive Least Square and Multi-Innovation Unscented Kalman Filter Algorithm. *Energies*. 2022. vol. 15. no. 4. DOI: 10.3390/en15041529.

38. Huang J., Zhu Z., Guo F., Huang G. The Devil Is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020. pp. 5699–5708. DOI: 10.1109/CVPR42600.2020.00574.
39. Lyu C., Zhang W., Huang H. et al. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. 2024. arXiv preprint arXiv:2212.07784.
40. Wan E. A., Van Der Merwe R. The unscented Kalman filter for nonlinear estimation. Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373). 2000. pp. 153–158. DOI: 10.1109/ASSPCC.2000.882463.
41. Volynsky M.A., Gurov I.P., Skakov P.S. [The recurrent algorithm for interferometric signals processing based on multi-cloud prediction model]. Nauchno-tehnicheskij vestnik informacionnyh tehnologij, mehaniki i optiki –Scientific and Technical Journal of Information Technologies, Mechanics and Optics. 2014. vol. 14. no. 4. pp. 18–22. (In Russ.).

Krasnov Dmitrii — Ph.D. student, Technical vision laboratory, ITMO University. Research interests: computer vision, pattern recognition, image processing, artificial intelligence, deep learning. The number of publications — 7. dmitriy_krasnov@outlook.com; 49, Kronverksky Av., 197101, St. Petersburg, Russia; office phone: +7(996)189-3418.

Volynsky Maxim — Ph.D., Associate Professor, Director, Technical vision laboratory, ITMO University. Research interests: computational photonics, biophotonics, interaction of optical radiation with matter, stochastic processing of quasi-periodic signals. The number of publications — 130. maxim.volynsky@gmail.com; 49, Kronverksky Av., 197101, St. Petersburg, Russia; office phone: +7(921)188-7946.

Gusev Alexandr — Ph.D., Lead engineer, Technical vision laboratory, ITMO University. Research interests: machine learning, pattern recognition, statistical learning, statistical pattern recognition. The number of publications — 32. gusew@internet.ru; 49, Kronverksky Av., 197101, St. Petersburg, Russia; office phone: +7(902)406-3854.

N.V. HUNG, T.T. HUONG, N. TAN, T.C. DOAN, N.N. HOANG
**MACHINE LEARNING APPLICATIONS FOR DELIVERY TIME
PREDICTION AND FREIGHT PLANNING**

Nguyen Viet Hung, Trinh Thu Huong, Nguyen Tan, Truong Cong Doan, Nguyen Nam-Hoang **Machine Learning Applications for Delivery Time Prediction and Freight Planning.**

Abstract. The rapid advancement of technology has a profound impact on logistics and freight transportation. Efficient management of transportation schedules is vital for businesses seeking to minimize costs, reduce delivery delays, and improve customer satisfaction. One of the most important challenges in this field is the Vehicle Routing Problem with Time Windows (VRPTW), which requires not only finding optimal delivery routes but also adhering to specific timing constraints for each customer or delivery point. Traditional optimization methods often struggle with the complexity and dynamic nature of real-world logistics, particularly when dealing with large-scale datasets and unpredictable factors such as traffic congestion or weather conditions. To address these limitations, this study introduces a machine learning-based system that enhances the performance of existing VRPTW solutions. Unlike conventional approaches that rely solely on heuristics or static planning, our system employs modern machine learning models to predict key time-related parameters – including transit time, availability time, and service time – based on historical and contextual data. These predictive capabilities allow the routing algorithms to make more informed decisions, resulting in more accurate and adaptable scheduling. Building on previous research involving Random Forest models, we propose a more robust framework that incorporates advanced preprocessing techniques and feature engineering to improve model accuracy. By training and evaluating the system using real-world datasets, we are able to simulate practical scenarios and validate the effectiveness of our approach. Experimental results show that our proposed method consistently outperforms other commonly used machine learning models in terms of Mean Absolute Error (MAE), thus confirming its potential for real-world applications. Overall, this study contributes a scalable and intelligent solution to a longstanding logistics problem, paving the way for more responsive and cost-effective transportation systems.

Keywords: Vehicle Routing Problem with Time Windows (VRPTW), machine learning models, logistics optimization, transit time prediction, random forest improvement, data processing techniques.

1. Introduction. The transportation industry plays a pivotal role in the global supply chain, especially as the volume of freight movement continues to surge in response to rising consumer demand and evolving market expectations. With the rapid growth of e-commerce, same-day delivery services, and global trade expansion, transportation systems are under increasing pressure to operate with high levels of precision and reliability. This intensifying demand places a significant burden on logistics providers, who must now balance efficiency, cost-effectiveness, and punctuality while navigating various real-world constraints such as traffic congestion, driver availability, and unpredictable weather conditions. As a result, transportation companies are continuously seeking innovative solutions to maintain their competitive edge in a highly dynamic and time-sensitive environment.

Among the most pressing challenges is the ability to meet strict delivery time requirements – a factor that directly impacts customer satisfaction and loyalty. Inaccurate delivery estimates can lead to missed time windows, increased operational costs, and reputational damage. To address these concerns, predictive analytics and optimization techniques have become essential tools. Specifically, the Vehicle Routing Problem with Time Windows (VRPTW) has emerged as a critical area of focus within logistics and operations research. VRPTW involves not only determining the most efficient delivery routes but also ensuring that deliveries occur within specified time frames for each customer. Solving this problem effectively requires sophisticated algorithms and predictive models capable of handling complex constraints. Consequently, researchers and industry professionals are increasingly leveraging machine learning and data-driven approaches to improve the accuracy of delivery time predictions, optimize routing strategies, and ultimately enhance the overall performance of freight transportation systems [1–4].

The Vehicle Routing Problem with Time Windows (VRPTW) is a well-defined optimization framework aimed at identifying efficient delivery routes while adhering to both spatial and temporal constraints [4–6]. The spatial component of VRPTW involves calculating actual distances between delivery points using geographic coordinates such as latitude and longitude. In parallel, the temporal dimension encompasses constraints related to delivery time windows and customer availability, which must be carefully respected to maintain service reliability. Effective solutions to VRPTW are critical not only for minimizing delivery times but also for reducing operational costs, enhancing fleet and personnel utilization, and satisfying increasingly diverse customer demands.

Recent advancements in artificial intelligence, particularly in machine learning, have provided powerful tools to address the complexity of VRPTW. Deep learning architectures such as Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Gated Recurrent Units (GRU) have proven effective in extracting temporal and spatial patterns from large-scale logistics data [7–9]. Additionally, ensemble-based models like Random Forest continue to be favored for their robustness and ability to handle heterogeneous data with high accuracy. These models are increasingly employed to predict critical time-related parameters – such as transport time, availability time, and service time – thereby enabling more accurate and adaptive route planning.

Despite their potential, the success of machine learning deployments in VRPTW heavily depends on the quality of the training data. Real-world

logistics data are often noisy, incomplete, or inconsistently structured, which can hinder model performance if not properly addressed. Therefore, effective data preprocessing techniques – such as data cleaning, normalization, feature engineering, and outlier removal – are essential to improve both the reliability and accuracy of predictive models. Enhancing data quality not only improves computational outcomes but also strengthens the practical applicability of intelligent routing systems in dynamic operational environments.

In this study, we place strong emphasis on the role of data preprocessing as a foundational step toward constructing a high-quality training dataset that enhances the effectiveness of machine learning models. Recognizing the limitations of existing approaches, we introduce an improved framework built upon the Random Forest framework, which leverages the strengths of ensemble learning to deliver more robust and interpretable results. While deep learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) architectures have shown great promise in various predictive tasks, our framework offers distinct advantages in terms of computational stability, efficiency, and the ability to handle high-dimensional and heterogeneous logistics data with minimal tuning.

The experimental evaluation of our framework reveals significant improvements in the accuracy of delivery time predictions, underscoring both its predictive power and its applicability in real-world scenarios. Through a systematic analysis of data preprocessing techniques – such as feature selection, normalization, and data cleaning – we demonstrate how enhancing dataset quality contributes directly to better model performance. Furthermore, by benchmarking our model against standard evaluation metrics and comparing it to existing machine learning approaches, we provide a comprehensive perspective on its practical advantages. Ultimately, this study presents a scalable and reliable solution to the Vehicle Routing Problem with Time Windows (VRPTW), contributing valuable insights to the ongoing efforts in logistics optimization within the modern transportation sector.

The remainder of this paper is organized as follows: Section 2 reviews related work on machine learning and optimization techniques for VRPTW. Section 3 presents the proposed methodology, including system design, data preprocessing, and feature construction. Section 4 describes the model training process and compares the proposed approach with various deep learning architectures. Section 5 provides a detailed performance evaluation of all models. Finally, Section 6 concludes the study and discusses future research directions.

2. Related work. The Vehicle Routing Problem with Time Windows (VRPTW) has been widely studied due to its practical importance in real-

world logistics and transportation systems. Traditional approaches to solving VRPTW mainly rely on exact algorithms (e.g., branch-and-bound, branch-and-cut) and heuristics or metaheuristics such as Genetic Algorithms, Ant Colony Optimization, and Tabu Search [10–15]. While these methods are effective for small or medium-scale instances, they often struggle with scalability and adaptability in complex, real-time environments.

To address these limitations, recent research has increasingly focused on leveraging deep learning (DL) and machine learning (ML) models to enhance prediction capabilities and routing decisions under dynamic conditions. These models such as [7–9], particularly Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs), have shown promising results in capturing both spatial and temporal patterns in data, which are crucial for optimizing routes with strict time constraints.

CNNs are particularly useful for processing spatial information such as traffic density, road layout, and geographic coordinates. Studies such as [16–18], have demonstrated the effectiveness of CNNs in extracting relevant spatial features that influence routing outcomes. By transforming traffic data into grid-like representations, CNNs can identify local spatial patterns and support real-time decision-making in dynamic routing environments.

On the other hand, LSTM and GRU models have been widely applied to capture temporal dependencies inherent in sequential transportation data. These models are adept at modeling delivery schedules, vehicle movement sequences, and time-window constraints. For instance, [19, 20] show that LSTM networks significantly outperform traditional time-series models in predicting delivery time, availability windows, and service durations. The ability of LSTM and GRU to retain long-term dependencies makes them highly suitable for scenarios where previous routing decisions influence future performance.

Furthermore, hybrid architectures that combine CNNs for spatial representation and RNNs (LSTM/GRU) for temporal learning have gained popularity. These hybrid models benefit from the strengths of both types of networks and are particularly effective in multi-objective VRPTW problems, where factors such as delivery time, fuel consumption, and customer satisfaction must be balanced. For example, [21–24] integrate CNN and GRU in a single framework, achieving superior performance in both accuracy and generalization across diverse routing scenarios.

Another emerging direction in VRPTW research is the application of Deep Reinforcement Learning (DRL). Unlike supervised learning models that require labeled data, DRL models learn optimal routing strategies through

interaction with an environment. Notably, DRL approaches using Pointer Networks and Transformer-based architectures have been proposed in [25–27] to tackle combinatorial optimization tasks, including VRPTW. These methods have shown strong potential in adapting to dynamic changes in demand and constraints by learning policies that generalize across problem instances.

Additionally, Graph Neural Networks (GNNs) are increasingly being employed to model the topological structure of road networks. Since road networks naturally form graphs, GNNs offer a more intuitive and effective representation compared to grid-based or sequential models. Research in [7, 28–30] has shown that GNNs, when combined with temporal models like LSTM or CNN [31, 32], significantly improve performance in routing tasks by incorporating both spatial connectivity and historical behavior.

Overall, the integration of deep learning models in solving VRPTW not only enhances prediction accuracy but also supports scalable and adaptive route planning systems. However, challenges remain in terms of model interpretability, real-time deployment, and generalization to unseen environments. Future work may focus on combining different deep learning paradigms (e.g., DRL with GNNs), incorporating domain knowledge through constraint-aware learning, and leveraging transfer learning to improve model robustness in varying logistical contexts.

3. Proposed Method

3.1. System Overview. This section presents the overall system architecture developed for delivery time prediction and freight planning based on the VRPTW framework. The proposed system consists of three main components: data preprocessing, model training, and route optimization. The workflow begins with transforming raw logistics data into structured input features, followed by training machine learning models to predict key delivery time parameters. These predictions are then integrated into route optimization algorithms to support effective decision-making. An overview of the system pipeline is illustrated in Figure 1.

The system consists of three main components:

- **Data Preprocessing.** The first step involves extracting and structuring VRPTW (Vehicle Routing Problem with Time Windows) data to create a training-ready dataset. This includes collecting data on customer locations, delivery time windows, vehicle capacities, and distances. The data is cleaned, normalized, and features are extracted, such as travel times, cluster analysis of delivery points, and distance metrics. This structured dataset serves as the foundation for training machine learning models.

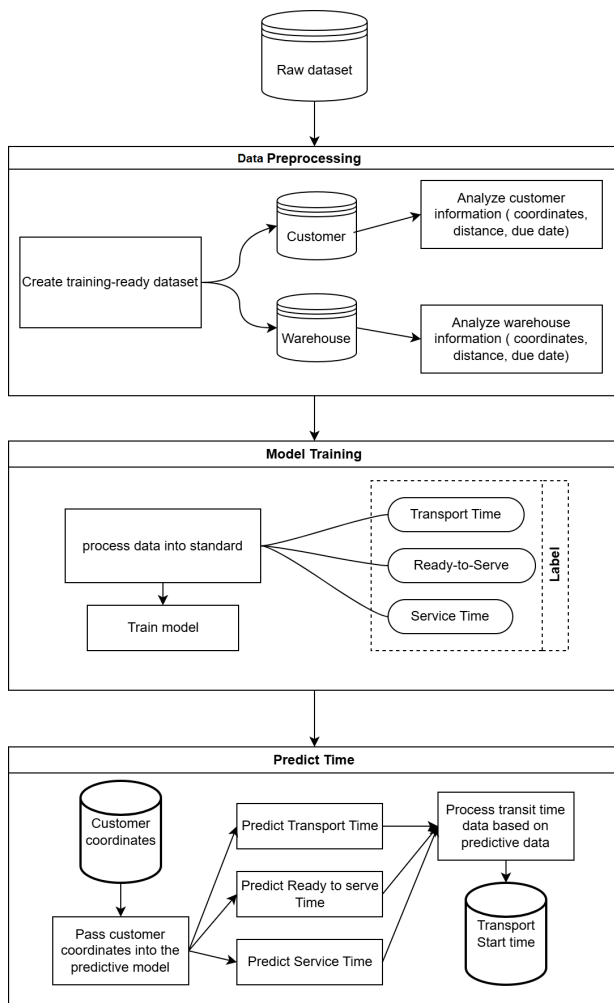


Fig. 1. Proposed System Architecture and Implementation Framework

– **Model Training.** In this phase, multiple machine learning models are trained to predict time metrics and optimize routes. Models include CNN (for spatial relationships), LSTM and GRU (for time-series predictions), and a proposed custom model designed for VRPTW optimization. The models are trained on the preprocessed dataset, with hyperparameter tuning and performance evaluation using metrics like accuracy and mean absolute

error (MAE). The goal is to identify the most effective model for real-world applications.

- **Route Optimization and Prediction.** Trained models are applied to optimize delivery routes and predict time metrics using real-world data. The system integrates machine learning predictions with classical optimization algorithms (e.g., Dijkstra, Genetic Algorithm) to calculate the shortest or most efficient routes while respecting constraints like time windows and vehicle capacities. The models also predict travel times based on real-time traffic and road conditions. Results are evaluated against actual delivery performance to ensure accuracy and reliability.

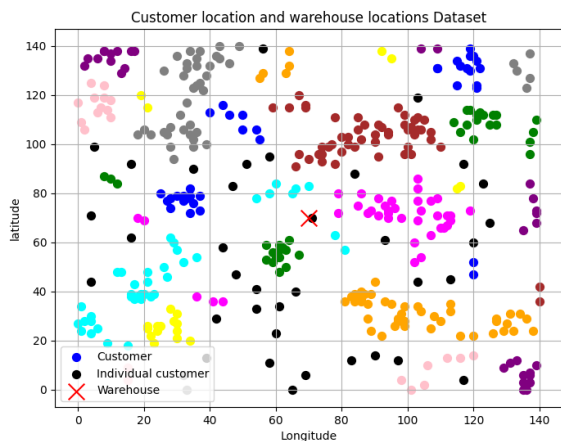
3.2. Building and Development Model

3.2.1. Data Source and Preprocessing. This research utilizes the **VRPTW Homberger dataset**, which includes instances with 100, 200, and 400 customers. It can be publicly accessed from: [33–35] Each instance contains three main components: **warehouse data**, which provides the spatial coordinates (x, y) of a single warehouse; **customer data**, which includes each customer's identifier, location (a, b) , demand, ready time, expiration time, and service time; and **vehicle data**, which specifies the number of vehicles and their maximum load capacity. To prepare the dataset for model training, we developed a **Data Preprocessing Module** to systematically extract, clean, and organize the data. This ensures the training data is reliable, minimizes potential risks, and improves the efficiency of downstream tasks.

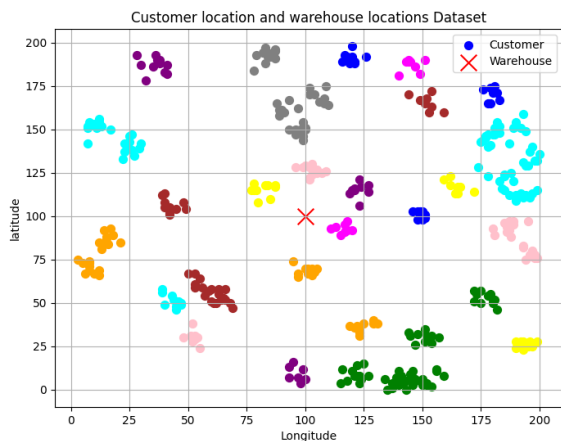
On the one hand, the preprocessing process begins with a detailed exploration of the dataset, which is categorized into three scales based on the number of customers: 100, 200, and 400. This classification allows the model to handle a variety of scenarios, from simple to complex. A thorough analysis confirms that the dataset provides all necessary information for computing input features, thereby reducing the risk of bias and improving data consistency. The next step involves extracting and organizing the data into three categories: **warehouse data**, which is duplicated to match the number of customers in each instance (e.g., repeated 400 times for a 400-customer dataset to facilitate pairwise calculations); **customer data**, which includes attributes such as demand, time windows, and location; and **vehicle data**, which specifies the number and capacity of vehicles.

On the other hand, Figures 2(a, b) illustrate the distribution of customer locations and warehouse sites within two distinct datasets.

In Figure 2(a), various colors represent individual customers, while the red cross marks the optimal warehouse location. The customer data points are scattered across a range of latitudes and longitudes, highlighting the geographic spread of customers.



a)



b)

Fig. 2. An example of clustering customers and optimal warehouse locations:
a) customer and warehouse location distribution – Dataset 1 (200 Customers);
b) customer and warehouse location distribution – Dataset 2 (400 Customers)

Figure 2(b) presents a similar analysis but features a different dataset, showcasing varying clusters of customers with the optimal warehouse location

again indicated by the red cross. Both visualizations provide insights into customer distribution patterns, which are essential for optimizing warehouse placement and improving logistics efficiency.

However, the central warehouse location was fixed in these examples for visualization purposes only. Although the warehouse appears centrally located, this does not represent an optimized warehouse position, nor does it assume uniform customer distribution. In fact, customer locations in these datasets are non-uniform, with varying densities and clustering patterns. These figures are intended to demonstrate spatial diversity and emphasize the model's ability to adapt to different geographic layouts. During training, the machine learning model does not rely on spatial centrality and is designed to generalize across varied logistical scenarios.

Finally, the extracted data is restructured into a **list of dictionaries**, where each dictionary represents a complete dataset. This structure separates the warehouse data (repeated for each customer) and customer data (containing detailed customer information), ensuring consistency and simplifying the definition of model inputs and outputs. By following this systematic approach, the data is prepared in a reliable and structured manner, ready to support efficient and flexible model training.

3.2.2. Feature Construction and Label Calculation. Upon completing the data preprocessing process, the next step involves the construction of features and the calculation of labels to create a structured input dataset suitable for training the machine learning model. These features and labels are derived based on the spatial, temporal, and operational constraints of the VRPTW problem.

The feature extraction process begins by pairing the coordinates of the warehouse and each customer to determine the spatial relationship between them. The Euclidean distance [36] between the warehouse and each customer is calculated using the formula proposed by Solomon [37], shown below:

$$D_i = \sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}, \quad (1)$$

where:

- D_i – Euclidean distance (in kilometers) between the warehouse and customer i .
- X_i, Y_i – Coordinates of customer i .
- X_k, Y_k – Coordinates of the warehouse.

In addition to the spatial features, temporal features play a critical role in capturing service constraints. Each customer's time window, defined by the

ready time (T_s) and expiration time (T_h), is included as part of the temporal features. The final feature set consists of the warehouse coordinates (X_k, Y_k), customer coordinates (X_i, Y_i), the distance (D_i), and the expiration time (T_h).

To simulate real-world operations, the label calculation process is performed concurrently with feature construction. The transportation time (T_v) from the warehouse to the customer is computed based on the Euclidean distance (D_i) and an assumed vehicle speed of 80 km/h. The transit time is converted to minutes using the following formula:

$$T_v = \frac{D_i}{80} \cdot 60, \quad (2)$$

where:

- T_v – Transit time (in minutes).
- D_i – Distance (in kilometers).

The availability time of each customer is adjusted to ensure compliance with operational constraints. Specifically, the adjusted service availability time (T_{sp}) is calculated by comparing the initial ready time (T_s) with a minimum threshold of 390 minutes (corresponding to 6:30 AM):

$$T_{sp} = \max(T_s, 390), \quad (3)$$

where:

- T_{sp} – Adjusted service availability time (in minutes).
- T_s – Initial ready time.

The service time at the customer location is then determined based on the difference between the expiration time (T_h) and the adjusted availability time (T_{sp}), with a maximum service time limit of 90 minutes to reflect operational constraints:

$$T_p = \min(T_h - T_{sp}, 90), \quad (4)$$

where:

- T_p – Service time (in minutes).
- T_h – Expiration time.
- T_{sp} – Adjusted service availability time.

After calculating features and labels, the dataset is thoroughly cleaned to ensure data quality. Records with invalid values, such as negative transit times or service times, are removed. This cleaning step eliminates inconsistencies and ensures all features and labels are valid and realistic.

The cleaned data is then organized into a structured data table, where features and labels are compiled and ready for further processing. To prevent potential order bias, the dataset is randomly shuffled before being split into training and testing sets, with 80% used for training and 20% for testing. Finally, all input features are normalized to the range $[0, 1]$ to enhance the training performance of the machine learning model.

It is important to note that the use of Euclidean distance in this study is a simplifying assumption. The VRPTW Homberger dataset does not provide road network data or graph-based routing information. As such, the calculated distance between the warehouse and each customer is based on direct linear distance (as per Solomon's formula [34]). This does not account for actual driving routes, road curvature, traffic constraints, or urban infrastructure. While this approach allows consistent feature engineering within the constraints of the dataset, we acknowledge it limits real-world accuracy. We have addressed this limitation in the conclusion and recommend future extensions to incorporate road network data or graph-based shortest path algorithms.

3.2.3. Pre-training Dataset. For a Homberger instance with 400 customers, the pre-training dataset is structured as follows:

- **Total Records.** The dataset contains a total of 400 records, with each record corresponding to an individual customer.
- **Features.** Each record includes 6 feature dimensions:
 - Warehouse coordinates (X_k, Y_k) .
 - Customer coordinates (X_i, Y_i) .
 - Euclidean distance (D_i) between the warehouse and the customer.
 - Customer expiration time (T_h) .
- **Labels.** Each record is associated with 3 label dimensions:
 - Transportation time (T_v) .
 - Adjusted service availability time (T_{sp}) .
 - Service time (T_p) .
- **Dataset Splitting.**
- **Training Set.** Consists of 320 records (80% of the dataset).
- **Test Set.** Comprises 80 records (20% of the dataset).

The dataset was manually split into 80% training and 20% testing by the authors. Prior to splitting, the data was randomly shuffled to minimize order bias and ensure generalizability. The structured dataset is saved in a reusable format, enabling further analysis and experimentation. This organization ensures the

dataset can be efficiently integrated into machine learning workflows while maintaining consistency and traceability for future use.

3.2.4. Preprocessed Data Analysis. The preprocessed dataset was analyzed to evaluate the feature distribution, label characteristics, and overall data quality, ensuring its suitability for training machine learning models.

Feature Distribution. The Euclidean distances exhibit high variability, reflecting the diverse geographic distribution of customers in the Homberger dataset. To prevent this variability from biasing model training, normalization was applied to scale the distances to a consistent range.

In addition, in the competitive landscape of logistics and transportation, accurately predicting delivery times is essential for optimizing operations and enhancing customer satisfaction. To achieve this, several key features must be considered during the modeling process. These features provide valuable insights into the factors that influence transportation efficiency and service delivery. Below, we delve into three critical label features that play a pivotal role in our predictive models:

- **Transit Time:** strongly correlated with the Euclidean distance, transit time provides a clear and predictable signal, making it an important feature for learning models.

- **Service Availability Time:** typically adjusted to a lower bound of 390 minutes (6:30 AM), this label captures operational constraints that models need to account for during prediction.

- **Service Time:** highly variable and capped at a maximum of 90 minutes, service time is the most challenging metric to predict due to its dependence on customer-specific constraints.

Data Quality. Preprocessing ensures that the dataset contains no missing or invalid values. The shuffling step effectively minimizes any order-related biases that may have been present in the original VRPTW files, thereby enhancing the randomness and robustness of the dataset.

Conclusion. This rigorous preprocessing and analysis process produces a high-quality dataset that is well-suited for training machine learning models. The standardized format and balanced feature-label representation ensure effective and unbiased model learning.

4. Model Training and Proposed Framework

4.1. The Importance of Predictive Labels. In this part, we focus on predicting three key delivery time-related labels, including:

Transit Time (T_v). This is the time it takes for a vehicle to travel from the warehouse to the customer, calculated based on the distance and speed of the vehicle. This label is important because it directly affects the total delivery time and the ability to plan routes efficiently. Accurately predicting the transit

time helps ensure that vehicles arrive on time, avoiding violations of customer time windows.

Ready to Serve Time (T_{sp}). This is the time when the customer is ready to receive the goods, adjusted to ensure that it is consistent with operational reality (minimum 390 minutes, corresponding to 6:30 AM). This label plays an important role in customer service sequencing, ensuring that vehicles arrive at the right time for customers to pick up their goods, thereby optimizing delivery schedules and reducing waiting times.

Service Time (T_p). This is the time required to deliver goods and complete related tasks at the customer's location, limited to a maximum of 90 minutes. Accurate prediction of service time helps ensure that the total dwell time at each customer does not exceed the allowed time window, and also supports planning so that vehicles can continue their route without delay.

Accurate prediction of these three labels is the foundation for optimizing delivery routes in the VRPTW problem. The above labels not only help to minimize operating costs but also improve customer satisfaction through on-time delivery. Therefore, selecting the appropriate machine learning model to predict these labels is an important task, requiring careful consideration between accuracy, generalizability, and computational efficiency.

4.2. Description and Reason for Choosing CNN, LSTM, GRU Models for Comparison. To evaluate the performance of the proposed model, we choose three popular deep learning models: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). These models are chosen because they represent state-of-the-art deep learning methods, are capable of handling complex data types and problems, and are widely used in time prediction research. Below is a detailed description and rationale for choosing each model:

4.2.1. Convolutional Neural Network (CNN). CNN is a type of deep learning neural network commonly used to process spatially structured data, such as images in [7]. In our research, we designed a CNN with 3 convolutional layers (Conv2D) with 32, 64, and 128 filters, combined with batch normalization layers to stabilize the training process, a max-pooling layer to reduce the feature dimension, and dense layers with 256, 128, and 3 neurons to make the final prediction. The input data is reformatted into a square image (e.g., 3x3x1) to match the CNN architecture.

We chose CNN because of its outstanding ability to extract spatial patterns from the data. Although the VRPTW data is largely tabular, reformatting the data into images allows the CNN to exploit spatial relationships between features such as warehouse coordinates, customer coordinates, and distances. CNNs are typically effective in prediction problems based on

spatially structured data, so they are a strong candidate for comparison with the proposed model to assess whether sophisticated deep learning methods can outperform traditional methods in this problem.

4.2.2. Long Short-Term Memory (LSTM). LSTM [8] is a variant of a recurrent neural network (RNN), designed to process sequence data and has the ability to remember information over long time periods thanks to the gate mechanism. In our research, LSTM consists of 2 LSTM layers with 128 and 64 units, in which the first layer has a `return_sequences` parameter (with the value `True`) to return the output sequence, combined with a dense layer (32 and 3 neurons) to predict three labels. The input data is reformatted into a sequence with 1 time step and 6 features to match the LSTM architecture.

LSTM is chosen because of its ability to handle temporal relationships, which may exist in VRPTW data through features such as ready time and expiration time. Although VRPTW data is not purely a time series, reformatting the data into a sequence allows LSTM to exploit latent time-related patterns. LSTM is commonly used in delivery time prediction or scheduling problems due to its ability to learn long-term dependencies, so it is a suitable model for comparison, to test whether sequence-based deep learning methods can provide an advantage over the proposed model.

However, we recognize that our implementation uses only a single time step per sample, which prevents the model from learning any true temporal dependencies. Additionally, in our PyTorch implementation, the hidden state is reset at the beginning of each batch, meaning that memory from previous sequences is not retained. As such, this model is not expected to outperform simpler architectures in this context, but was included to test the behavior of gated memory units when applied to static inputs.

4.2.3. Gated Recurrent Units (GRU). GRU [9] is another variant of RNN, similar to LSTM but simpler by using fewer gates (update and reset gates). The GRU model in this research consists of 2 GRU layers, each with 64 units, with the first layer having the `return_sequences` parameter set to `True`, combined with batch normalization and dense layers (128 and 3 neurons) to make predictions. The input data is also formatted as a sequence similar to LSTM.

We chose GRU because it is a lighter version of LSTM, providing higher computational efficiency while maintaining the ability to learn temporal dependencies. GRU is often used in problems that require sequence processing but require faster training speed than LSTM, such as time prediction in logistics. Comparing GRU with the proposed model helps to evaluate whether a lighter deep learning model can compete with the traditional method, and also provides

a comprehensive view of the performance of deep learning methods on this problem.

As with LSTM, the GRU in this study processes only one time step per sample. Because the hidden state is reinitialized at each batch, the network cannot benefit from sequence memory. This limits the ability of GRU to learn meaningful temporal patterns in our setup, but allows us to evaluate its effectiveness when applied to tabular data through a sequence-like structure.

4.3. Advantages of our framework and Model Structure Comparison. Our proposal can be called Delivery Time Prediction – Machine Learning (DTP-ML).

Description. Our framework is a Random Forest-based model [38], an ensemble learning method that uses multiple training iterations to make predictions. The DTP-ML model is designed with 90–100 training iterations (the optimal range determined by the experiment shown in Table 1), the input data is tabular data with 6 features and does not require data reformatting. We have taken advantage of the voting mechanism of training iterations to increase accuracy and reduce overfitting.

Table 1. Model structures, inputs, and computational requirements

Model	Main Structure	Input	Number of Main Layers	Tuning Parameters	Computational Requirements
CNN	3 Conv2D layers (32, 64, 128 filters), MaxPooling, Dense (256, 128, 3)	Square Image (3x3x1)	3 convolutional layers, 3 dense layers	Learning rate (0.001), dropout rate (0.001), number of filters	High (requires GPU)
LSTM	2 LSTM layers (128, 64 units), Dense (32, 3)	Chain (1 time step, 6 features)	2 LSTM layers, 2 dense layers	Learning rate (0.001), dropout rate (0.001), number of units	Very High (requires GPU)
GRU	2 GRU layers (64 units), Dense (128, 3)	Chain (1 time step, 6 features)	2 GRU layers, 2 dense layers	Learning rate (0.001), dropout rate (0.001), number of units	High (requires GPU)
DTP-ML	Training batch (90–100 trainings)	Panel data (6 features)	Multiple training	Number of training runs (90–100)	Low (no GPU required)

4.3.1. Advantages compared to CNN, LSTM, GRU. Suitable for Tabular Data [39]: VRPTW data is tabular, and the proposed model has directly processed this data without reformatting, avoiding noise or information loss as in CNN (must be formatted into images) or LSTM/GRU (must be formatted into sequences). This allows the DTP-ML model to exploit the maximum information from features such as distance and coordinates.

4.3.2. Stability and Generalization. We applied an ensemble learning mechanism to DTP-ML, combining predictions from multiple training runs to

reduce overfitting. Meanwhile, deep learning models such as CNN, LSTM, and GRU are prone to overfitting if not carefully tuned (e.g. needing a dropout rate of 0.001 and stopping early), especially with small or unstructured data.

4.3.3. Computational Efficiency. Our proposed model has a significantly faster training time than deep learning models, which require GPU resources and many iterations (150 rounds). With 90–100 training runs, DTP-ML achieves high accuracy without the need for powerful hardware, suitable for real-time applications in logistics.

4.3.4. Easy Tuning. DTP-ML only requires tuning the number of training iterations (90–100 is optimal), while CNN, LSTM, and GRU require tuning many parameters such as learning rate, dropout rate, number of layers, and number of units in each layer. This simplicity makes the model easier to deploy in real-world scenarios and easier to maintain.

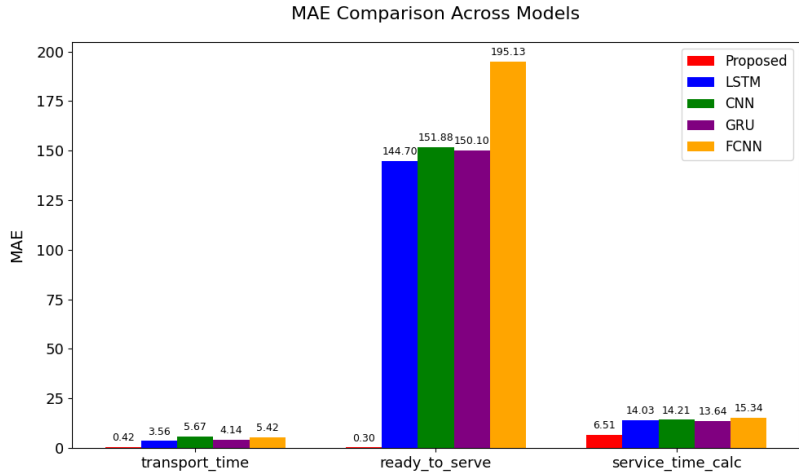
On the other hand, we can see that in Table 1 shows that DTP-ML has a simpler structure, does not require complex data block reformatting, and has significantly lower computational requirements than CNN, LSTM, and GRU. This highlights the superiority of the proposed model in predicting delivery time, especially in the context of the VRPTW problem with structured table data.

While our proposed model leverages ensemble learning for tabular data, we recognize that other architectures designed specifically for tabular regression problems – such as FT-Transformer [40] and GLN [41] – offer promising directions. These models are capable of modeling non-linear interactions in high-dimensional data with strong regularization. Although not implemented in this study due to computational constraints, we propose incorporating these architectures in future work to further benchmark our solution.

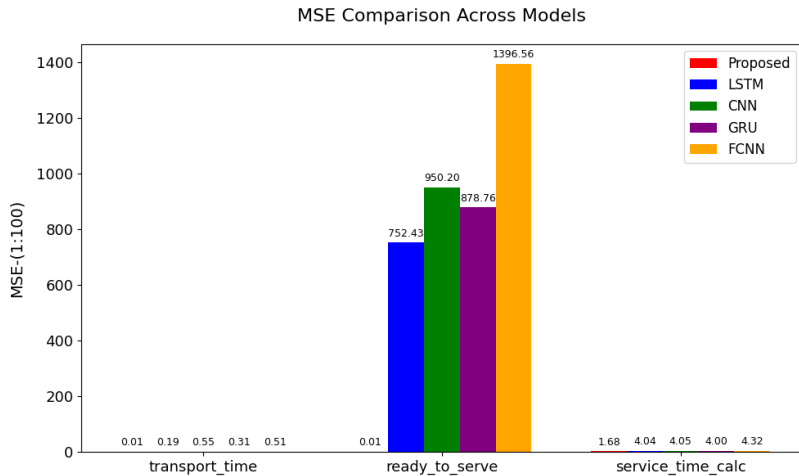
We also acknowledge that a two-layer fully connected neural network (FCNN) is a relevant and commonly used baseline for tabular regression problems. We plan to include this model in future extensions of our research to further enrich the performance comparison and establish more representative benchmarks.

5. Performance Evaluation

5.1. Training performance. Figure 3 and Figure 4 shows the performance of all models on the test set, using the metrics Mean Absolute Error (MAE) in the formula 5, Mean Squared Error (MSE) in the formula 6, Coefficient of Determination (R^2) in the formula (7), and Accuracy (relative error < 10%) in the formula (8).



a)



b)

Fig. 3. Error Metric Comparison (MAE and MSE) of machine learning models for delivery time prediction: a) mean Absolute Error (MAE) of different models on delivery time features; b) mean Squared Error (MSE) comparison among models

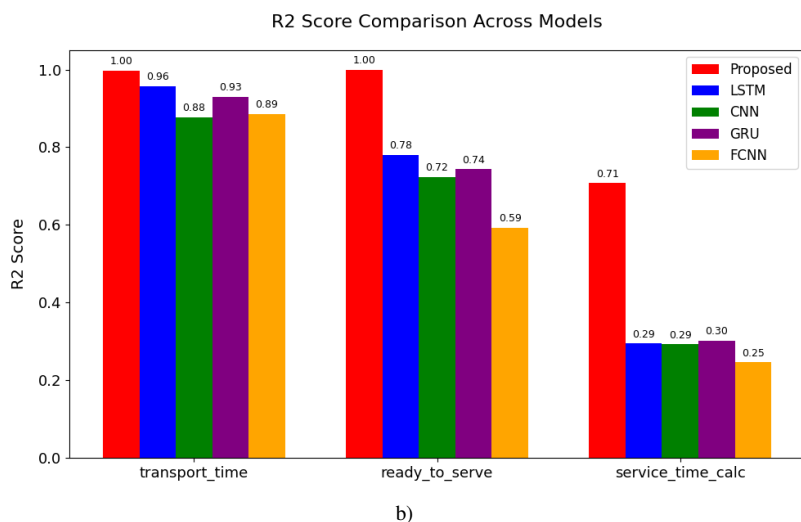
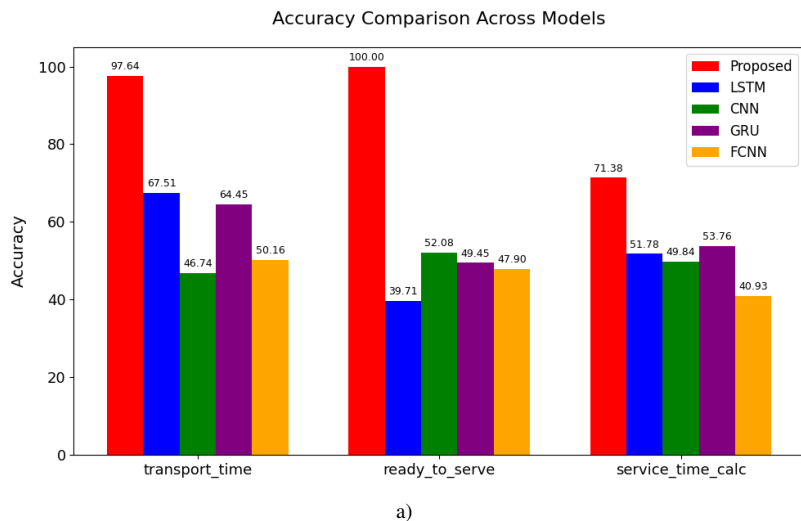


Fig. 4. Accuracy and R^2 score comparison of machine learning models for delivery time prediction: a) accuracy comparison of proposed, LSTM, CNN, and GRU models on delivery time features; b) R^2 Score evaluation of all models across delivery time features

On the one hand, before evaluating the model's performance, the following metrics were used to quantitatively assess the prediction results:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i^{label} - \hat{y}_i^{label}|, \quad (5)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i^{label} - \hat{y}_i^{label})^2, \quad (6)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i^{label} - \hat{y}_i^{label})^2}{\sum_{i=1}^n (y_i^{label} - \bar{y}^{label})^2}, \quad (7)$$

$$Accuracy = \left(\frac{\sum_{i=1}^n \mathbb{I} \left(\frac{|y_i^{label} - \hat{y}_i^{label}|}{y_i^{label}} < 0.1 \right)}{n} \right) \times 100\%. \quad (8)$$

In the formulas above:

- y_i^{label} represents the true value of the target variable for sample i . In this research, the target variables are: Transport time, Ready to serve, or Service time.
- $\hat{y}_i^{label}$ denotes the predicted value generated by the model for sample i .
- $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition inside is true (i.e., the relative error between the predicted value and the true value is less than 10%), and returns 0 otherwise.

On the other hand, the **Transport Time Prediction** that in the task of predicting transport time, the DTP-ML model achieved the best performance with a Mean Absolute Error (MAE) of 0.4203, outperforming the LSTM model, which recorded a slightly higher MAE of 0.6659. This indicates the superior precision of DTP-ML in modeling this task. In contrast, CNN and GRU exhibited significantly poorer performance, with MAE values of 2.7542 and 2.9459, respectively. In terms of accuracy, DTP-ML achieved a high score of 97.64%, reinforcing its effectiveness in this task.

In addition, the **Ready-to-Serve Time Prediction** for the ready-to-serve time task, both DTP-ML and LSTM models reached perfect prediction accuracy (100%) and R^2 scores of 1.0000, suggesting an excellent fit to the data.

However, DTP-ML clearly outperformed LSTM in terms of error magnitude, achieving a much lower MAE of 0.3031 compared to LSTM's 1.4463. On the other hand, CNN and GRU exhibited substantially higher MAE values of 15.0827 and 13.4415, respectively. Although their R^2 values remained high, this is likely due to the constrained distribution of the labels rather than true predictive accuracy.

Finally, the **Service Time Prediction**, in the most challenging task – service time prediction – DTP-ML once again demonstrated its superiority. It achieved an MAE of 8.5142 and an R^2 of 0.7075, which are significantly better than those of the other models. LSTM, CNN, and GRU showed much higher MAE values (14.9116, 15.6824, and 15.8123, respectively) and correspondingly low R^2 scores (ranging from 0.1206 to 0.2066), indicating weaker model fit. DTP-ML also attained the highest prediction accuracy for this task, with a score of 71.38%, whereas all other models recorded accuracy rates below 46%.

5.2. Effect of Training Times in DTP-ML. Table 2 presents the accuracy (in percentage) of the DTP-ML model across three tasks – Transport Time, Ready-to-Serve Time, and Service Time – under varying numbers of training iterations, ranging from 50 to 150. To better understand the model's performance stability and trends across these tasks, a detailed analysis and evaluation are conducted based on the results shown in the table.

Table 2. Accuracy for different numbers of experimental training

No. of Times	Transport Times (%)	Ready-to-Serve Times (%)	Service Times (%)
50	97.67	82.62	81.68
60	97.67	82.92	81.68
70	97.72	83.02	81.77
80	97.73	82.85	81.63
90	97.76	82.74	81.65
100	97.88	82.63	81.74
110	97.76	82.62	81.85
120	97.84	82.72	81.67
130	97.75	82.71	81.65
140	97.78	82.69	81.65
150	97.82	82.75	81.65

5.2.1. Delivery Time. Accuracy increases gradually from 97.67% (50 training runs) to a peak of 97.88% (100 training runs), and then fluctuates slightly between 97.75–97.84% as the number of training runs increases to 150. This shows that the performance of DTP-ML on this task peaks at 100 training

runs and stabilizes thereafter, confirming that around 90–100 training runs is the optimal choice to achieve high accuracy without increasing computational complexity.

5.2.2. Ready to Serve Time. Accuracy fluctuates between 82.62–83.02%, with the highest value achieved at 70 training runs (83.02%). However, after 70 training runs, the accuracy drops slightly and fluctuates around 82.62–82.75%. This may reflect that higher training runs do not provide a significant benefit for this task, and a model with 70–100 training runs is sufficient to achieve good performance. Although the accuracy in this table is lower than the 100% recorded in Table 1, this may be due to factors such as the way the accuracy is calculated or the different test datasets.

5.2.3. Service Time. Accuracy ranges from 81.63% to 81.85%, with the highest value achieved at 110 training runs (81.85%). However, the improvement between training runs is very small, and the accuracy stabilizes around 81.65–81.85% from 90 training runs onwards. This suggests that the performance of DTP-ML for this task also reaches a steady state around 90–100 training runs, similar to other tasks.

5.2.4. Overall Comments. Based on the accuracy table, DTP-ML shows stable and peak performance with training runs from 90 to 100 on all three tasks. The highest accuracy for Transit Time (97.88%) and the stability in other tasks (around 82–83% for Ready to Serve Time and 81–82% for Service Time) confirm that DTP-ML does not need too many training runs to achieve optimal performance. The choice of 90–100 training runs is reasonable, balancing accuracy and computational efficiency, which is in line with the requirements of real logistics applications.

6. Conclusions. Our research introduces a machine learning-based framework for delivery time prediction, focusing on three critical time-related indicators in logistics operations: Transport Time, Ready-to-Serve Time, and Service Time. Through rigorous preprocessing – including feature engineering, label construction, and normalization – we prepared high-quality datasets using synthetic data from the VRPTW Homberger benchmark. To evaluate model performance across these tasks, we conducted a comprehensive set of experiments comparing several architectures, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and a customized machine learning training strategy (DTP-ML).

The data preprocessing pipeline played a crucial role in enhancing model effectiveness, involving feature extraction, label construction, and normalization techniques to ensure consistent input quality. Among the models evaluated, the proposed DTP-ML approach, based on Random Forests and trained across 90–100 iterations, consistently outperformed deep learning

counterparts. It achieved superior performance with MAE values of 0.4203 (Transport Time), 0.3031 (Ready-to-Serve Time), and 8.5142 (Service Time), along with high prediction accuracy. In addition to its low error rates, the DTP-ML model demonstrated strong robustness and computational efficiency, requiring minimal hardware resources – making it a highly viable solution for real-world logistics optimization tasks.

However, we acknowledge several important limitations of the current study. First, our model uses Euclidean distance as a proxy for transit time, which does not capture real-world road network constraints such as travel paths, intersections, or dynamic traffic conditions. This simplification was necessary due to the structure of the benchmark dataset, which does not include graph-based road information. Consequently, our results may be less accurate when applied directly to real-world urban environments unless integrated with actual geospatial data and routing graphs. Future research should consider extending the current framework by incorporating graph-based routing data or using models designed for spatial networks.

Second, we recognize that the deep learning models originally selected (CNN, LSTM, GRU) are not inherently optimized for tabular data. Their inclusion was intended to explore their adaptability to non-sequential inputs. To provide a fairer comparison, we have introduced a fully connected neural network (FCNN) as a more suitable deep learning baseline and discussed state-of-the-art tabular architectures such as FT-Transformer and GLN, which we propose as important directions for future benchmarking and development.

In summary, this study contributes a scalable, interpretable, and computationally efficient model for delivery time prediction in VRPTW scenarios, with clear application potential in freight planning. Future directions for this work include:

- Integrating road network data to replace Euclidean distance approximation.
- Benchmarking against advanced tabular architectures like FT-Transformer and GLN.
- Integrating hybrid models that combine the interpretability and efficiency of tree-based models (like DTP-ML) with the learning capacity of deep learning architectures (e.g., CNN, LSTM).
- Applying the framework to larger-scale, real-world logistics datasets to validate generalizability across different domains and geographic contexts.
- Exploring online learning capabilities for real-time prediction updates in dynamic environments.
- Incorporating geospatial and temporal data fusion to further enhance model performance under complex delivery conditions.

These future extensions aim to strengthen the practical value and adaptability of machine learning in the evolving landscape of smart logistics and intelligent transportation systems.

References

1. Minh N.N., Tien N.H. Factors affecting career opportunities abroad for students of the faculty of business administration of the HCMC university of food industry. *International Journal of Multidisciplinary Reseach and Growth Evaluation*. 2024. vol. 5. no. 1. pp. 556–565.
2. Li F., Yang X., Zhu R., Li T., Liu J. Growth mechanism in transformation and upgrading of logistics industry. *Systems*. 2025. vol. 13. no. 3. p. 202.
3. Hung N.V., Luu D.L., Dong N.S., Lupu C. Reducing time headway for cooperative vehicle following in platoon via information flow topology. *Journal of Control Engineering and Applied Informatics*. 2024. vol. 26. no. 2. pp. 77–87.
4. Maroof A., Ayvaz B., Naeem K. Logistics optimization using hybrid genetic algorithm (hga): a solution to the vehicle routing problem with time windows (vrptw). *IEEE Access*. 2024.
5. Leng K., Li S. Distribution path optimization for intelligent logistics vehicles of urban rail transportation using VRP optimization model. *IEEE Transactions on Intelligent Transportation Systems*. 2021. vol. 23. no. 2. pp. 1661–1669.
6. Luu D.L., Hung N.V., Lupu C. Tracking trajectory by using the polynomial method for acc system based on smart car platform. 17th International Conference on Engineering of Modern Electric Systems (EMES). IEEE, 2023. pp. 1–4.
7. Kovacs L., Jlidi A. Neural networks for vehicle routing problem. *arXiv preprint arXiv:2409.11290*. 2024.
8. Osman O., Rakha H., Mittal A. Application of long short term memory networks for long-and short-term bus travel time prediction. *Preprints*. 2021. vol. 2021040269.
9. Iklasov Z., Sobirov I., Solozabal R., Takac M. Reinforcement learning for solving stochastic vehicle routing problem with time windows. *arXiv preprint arXiv:2402.09765*. 2024.
10. Daouda A.S.M., Atila U. A hybrid particle swarm optimization with tabu search for optimizing aid distribution route. *Artificial Intelligence Studies*. 2024. vol. 7. no. 1. pp. 10–27.
11. Dewantoro R.W., Sihombing P. et al. The combination of ant colony optimization (aco) and tabu search (ts) algorithm to solve the traveling salesman problem (tsp). 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM). IEEE, 2019. pp. 160–164.
12. Rohini V., Natarajan A. Comparison of genetic algorithm with particle swarm optimisation, ant colony optimisation and tabu search based on university course scheduling system. *Indian Journal of Science and Technology*. 2016. vol. 9. no. 21. pp. 1–5.
13. Chen J., Gui P., Ding T., Na S., Zhou Y. Optimization of transportation routing problem for fresh food by improved ant colony algorithm based on tabu search. *Sustainability*. 2019. vol. 11. no. 23. p. 6584.
14. Zinov V., Kartak V., Valiakhmetova Y. Solving multi-objective rational placement of load-bearing walls problem via genetic algorithm. *Informatics and Automation*. 2025. vol. 24. no. 2. pp. 464–491.
15. Gharbi A., Ayari M., El Touati Y. Intelligent agent-controlled elevator system: Algorithm and efficiency optimization. *Informatics and Automation*. 2025. vol. 24. no. 1. pp. 30–50.

16. Nazari M., Oroojlooy A., Snyder L., Takac M. Reinforcement learning for solving the vehicle routing problem. *Advances in neural information processing systems*. 2018. vol. 31.
17. Wu R., Wang R., Hao J., Wu Q., Wang P., Niyato D. Multiobjective vehicle routing optimization with time windows: A hybrid approach using deep reinforcement learning and nsga-ii. *IEEE Transactions on Intelligent Transportation Systems*. 2024.
18. Ata K.I.M., Hassan M.K., Ismaeel A.G., Al-Haddad S.A.R., Alani S. et al. A multi-layer cnn-gruskip model based on transformer for spatial-temporal traffic flow prediction. *Ain Shams Engineering Journal*. 2024. vol. 15. no. 12. p. 103045.
19. Zhang X., Chan K.W., Li H., Wang H., Qiu J., Wang G. Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model. *IEEE transactions on cybernetics*. 2020. vol. 51. no. 6. pp. 3157–3170.
20. Yang Y.-Q., Lin J., Zheng Y.-B. Short-time traffic forecasting in tourist service areas based on a cnn and gru neural network. *Applied Sciences*. 2022. vol. 12. no. 18. p. 9114.
21. Guo F., Wei Q., Wang M., Guo Z., Wallace S.W. Deep attention models with dimension-reduction and gate mechanisms for solving practical time-dependent vehicle routing problems. *Transportation research part E: logistics and transportation review*. 2023. vol. 173. p. 103095.
22. Zafar N., Haq I.U., Chughtai J.-u.-R., Shafiq O. Applying hybrid lstm-gru model based on heterogeneous data sources for traffic speed prediction in urban areas. *Sensors*. 2022. vol. 22. no. 9. p. 3348.
23. Hou M., Liu S., Zheng Q., Liu C., Zhang X., Kang C. A deep learning based communication traffic prediction approach for smart monitoring of distributed energy resources in virtual power plants. *IET Smart Grid*. 2024. vol. 7. no. 5. pp. 653–671.
24. Hung N.V., Luu D.L., Pham Q.T., Lupu C. Comparative analysis of different spacing policies for longitudinal control in vehicle platooning. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. 2024. DOI: 10.1177/09544070241273985.
25. Kool W., Van Hoof H., Welling M. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*. 2018.
26. Cheng Q., Yuanyai C., Zhan Y.-Q., Cheng C.-Y. Optimizing vehicle routing by incorporating driver preferences: A pointer network and deep q network (pndqn) approach.
27. Li K., Zhang T., Wang R. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*. 2020. vol. 51. no. 6. pp. 3103–3114.
28. Jiang W., Han H., Zhang Y., Wang J., He M., Gu W., Mu J., Cheng X. Graph neural networks for routing optimization: Challenges and opportunities. *Sustainability*. 2024. vol. 16. no. 21. p. 9239. DOI: 10.3390/su16219239.
29. Bahovska E. Graph neural networks in neighbourhood selection for a vehicle routing problem solver. *Master's thesis*. 2023.
30. Tien Z.C., Qi-lee J. Enhancing vehicle routing problem solutions through deep reinforcement learning and graph neural networks. *International Journal of Enterprise Modelling*. 2022. vol. 16. no. 3. pp. 125–135.
31. Nguyen H., Dao T.N., Pham N.S., Dang T.L., Nguyen T.D., Truong T.H. An accurate viewpoint estimation method for 360 video streaming using deep learning. *EAI Endorsed Trans. Ind. Networks Intell. Syst*. 2022. vol. 9. no. 4. p. e2. DOI: 10.4108/eetinis.v9i4.2218.
32. Inzillo V., Garompolo D., Giglio C. Enhancing smart city connectivity: A multi-metric cnn-lstm beamforming based approach to optimize dynamic source routing in 6g networks for MANETs and VANETs. *Smart Cities*. 2024. vol. 7. no. 5. pp. 3022–3054.

33. “solomon-100”. Available at: <https://www.sintef.no/globalassets/project/top/vrptw/solomon/solomon-100.zip> (accessed 30.05.2025).
34. “solomon-200”. Available at: https://www.sintef.no/globalassets/project/top/vrptw/homberger/200/homberger_200_customer_instances.zip (accessed 30.05.2025).
35. “solomon-400”. Available at: https://www.sintef.no/globalassets/project/top/vrptw/homberger/400/homberger_400_customer_instances.zip (accessed 30.05.2025).
36. Weisstein E.W. Distance. html, 2003, from MathWorld – A Wolfram Web Resource. Available at: <https://mathworld.wolfram.com/Distance> (accessed 30.05.2025).
37. Solomon M.M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*. 1987. vol. 35. no. 2. pp. 166–324.
38. Kim Y.H., Kim I., Kim Y.-J., Kim M., Cho J.-H., Hong M., Kang K.-H., Lim S.-H., Kim S.-J., Kim N. et al. The prediction of sagittal chin point relapse following two-jaw surgery using machine learning. *Scientific Reports*. 2023. vol. 13. no. 1. DOI: 10.1038/s41598-023-44207-2.
39. Sulistio B., Warnars H.L.H.S., Gaol F.L., Soewito B. Energy sector stock price prediction using the CNN, GRU LSTM hybrid algorithm. *International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*. IEEE, 2023. pp. 178–182.
40. Gorishniy Y., Rubachev I., Khrulkov V., Babenko A. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*. 2021. vol. 34. pp. 18932–18943.
41. Popov S., Morozov S., Babenko A. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*. 2019.

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 — 32. hungnv@eaut.edu.vn; Ky Anh, Ha Tinh, Viet Nam; office phone: +84(098)911-2079.

Trinh Thu Huong — Lecturer, International training and cooperation institute, East Asia University of Technology. Research interests: user logistics and supply chain management, business administration, economics and management. The number of publications — 2. huongtt2@eaut.edu.vn; Thanh Hoa, Viet Nam; office phone: +84(090)309-4888.

Nguyen Tan — Research assistant, East Asia University of Technology. Research interests: applications, data analysis. The number of publications — 4. tan25102000@gmail.com; Hung Yen, Viet Nam; office phone: +84(035)919-0216.

Truong Cong Doan — Ph.D., Lecturer, International school, Vietnam National University Hanoi. Research interests: applied artificial intelligence, machine learning, and data science in finance, marketing, management. The number of publications — 30. tcdoan@vnu.edu.vn; Hanoi, Viet Nam; office phone: +84(090)215-4359.

Nguyen Nam-Hoang — Associate professor, Faculty of information technology, East Asia University of Technology. Research interests: mobile communications networks, Internet of Things, visible light communications. The number of publications — 47. hoangnguyen@eaut.edu.vn; Thanh Xuan, Hanoi, Viet Nam; office phone: +84(24)3754-9338.

Н.В. ХУНГ, Т. ТУ ХУОНГ, Н. ТАН, Т.К. ДОАН, Н. НАМ-ХОАНГ
**ПРИЛОЖЕНИЯ МАШИННОГО ОБУЧЕНИЯ ДЛЯ
ПРОГНОЗИРОВАНИЯ СРОКОВ ДОСТАВКИ И
ПЛАНИРОВАНИЯ ПЕРЕВОЗОК**

Хунг Н.В., Ту Хуонг Т., Тан Н., Доан Т.К., Нам-Хоанг Н. Приложения машинного обучения для прогнозирования сроков доставки и планирования перевозок.

Аннотация. Стремительное развитие технологий оказывает огромное влияние на логистику и грузоперевозки. Эффективное управление графиками перевозок критически важно для предприятий, стремящихся минимизировать затраты, сократить задержки доставки и повысить удовлетворенность клиентов. Одна из ключевых задач в этой области – задача маршрутизации транспортных средств с временными окнами (VRPTW), которая требует не только поиска оптимальных маршрутов доставки, но и соблюдения определенных временных ограничений для каждого клиента или пункта доставки. Традиционные методы оптимизации часто сталкиваются со сложностью и динамичностью реальных логистических процессов, особенно при работе с большими объемами данных и непредсказуемыми факторами, такими как пробки на дорогах или погодные условия. Для устранения этих ограничений в данном исследовании представлена система на основе машинного обучения, которая повышает производительность существующих решений VRPTW. В отличие от традиционных подходов, которые полагаются исключительно на эвристику или статическое планирование, наша система использует современные модели машинного обучения для прогнозирования ключевых временных параметров, включая время доставки, время доступности и время обслуживания, на основе исторических и контекстных данных. Эти возможности прогнозирования позволяют алгоритмам маршрутизации принимать более обоснованные решения, что приводит к более точному и адаптируемому планированию. Опираясь на предыдущие исследования с использованием моделей случайного леса, мы предлагаем более надежную структуру, которая включает в себя передовые методы предварительной обработки и проектирование признаков для повышения точности модели. Обучая и оценивая систему с использованием реальных наборов данных, мы можем моделировать практические сценарии и подтверждать эффективность нашего подхода. Результаты экспериментов показывают, что предложенный метод стабильно превосходит распространенные модели машинного обучения с точки зрения средней абсолютной ошибки (MAE), тем самым подтверждая его потенциал для практического применения. Таким образом, данное исследование вносит свой вклад в масштабируемое и интеллектуальное решение давней логистической проблемы, открывая путь к более гибким и экономически эффективным транспортным системам.

Ключевые слова: задача маршрутизации транспортных средств с временными окнами (VRPTW), модели машинного обучения, оптимизация логистики, прогнозирование времени в пути, улучшение моделей случайного леса, методы обработки данных.

Литература

1. Minh N.N., Tien N.H. Factors affecting career opportunities abroad for students of the faculty of business administration of the HCMC university of food industry. International Journal of Multidisciplinary Research and Growth Evaluation. 2024. vol. 5. no. 1. pp. 556–565.

2. Li F., Yang X., Zhu R., Li T., Liu J. Growth mechanism in transformation and upgrading of logistics industry. *Systems*. 2025. vol. 13. no. 3. p. 202.
3. Hung N.V., Luu D.L., Dong N.S., Lupu C. Reducing time headway for cooperative vehicle following in platoon via information flow topology. *Journal of Control Engineering and Applied Informatics*. 2024. vol. 26. no. 2. pp. 77–87.
4. Maroof A., Ayvaz B., Naeem K. Logistics optimization using hybrid genetic algorithm (hga): a solution to the vehicle routing problem with time windows (vrptw). *IEEE Access*. 2024.
5. Leng K., Li S. Distribution path optimization for intelligent logistics vehicles of urban rail transportation using VRP optimization model. *IEEE Transactions on Intelligent Transportation Systems*. 2021. vol. 23. no. 2. pp. 1661–1669.
6. Luu D.L., Hung N.V., Lupu C. Tracking trajectory by using the polynomial method for acc system based on smart car platform. 17th International Conference on Engineering of Modern Electric Systems (EMES). *IEEE*, 2023. pp. 1–4.
7. Kovacs L., Jlidi A. Neural networks for vehicle routing problem. *arXiv preprint arXiv:2409.11290*. 2024.
8. Osman O., Rakha H., Mittal A. Application of long short term memory networks for long-and short-term bus travel time prediction. *Preprints*. 2021. vol. 2021040269.
9. Iklassov Z., Sobirov I., Solozabal R., Takac M. Reinforcement learning for solving stochastic vehicle routing problem with time windows. *arXiv preprint arXiv:2402.09765*. 2024.
10. Daouda A.S.M., Atila U. A hybrid particle swarm optimization with tabu search for optimizing aid distribution route. *Artificial Intelligence Studies*. 2024. vol. 7. no. 1. pp. 10–27.
11. Dewantoro R.W., Sihombing P. et al. The combination of ant colony optimization (aco) and tabu search (ts) algorithm to solve the traveling salesman problem (tsp). 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM). *IEEE*, 2019. pp. 160–164.
12. Rohini V., Natarajan A. Comparison of genetic algorithm with particle swarm optimisation, ant colony optimisation and tabu search based on university course scheduling system. *Indian Journal of Science and Technology*. 2016. vol. 9. no. 21. pp. 1–5.
13. Chen J., Gui P., Ding T., Na S., Zhou Y. Optimization of transportation routing problem for fresh food by improved ant colony algorithm based on tabu search. *Sustainability*. 2019. vol. 11. no. 23. p. 6584.
14. Zinov V., Kartak V., Valiakhmetova Y. Solving multi-objective rational placement of load-bearing walls problem via genetic algorithm. *Informatics and Automation*. 2025. vol. 24. no. 2. pp. 464–491.
15. Gharbi A., Ayari M., El Touati Y. Intelligent agent-controlled elevator system: Algorithm and efficiency optimization. *Informatics and Automation*. 2025. vol. 24. no. 1. pp. 30–50.
16. Nazari M., Oroojlooy A., Snyder L., Takac M. Reinforcement learning for solving the vehicle routing problem. *Advances in neural information processing systems*. 2018. vol. 31.
17. Wu R., Wang R., Hao J., Wu Q., Wang P., Niyato D. Multiobjective vehicle routing optimization with time windows: A hybrid approach using deep reinforcement learning and nsga-ii. *IEEE Transactions on Intelligent Transportation Systems*. 2024.
18. Ata K.I.M., Hassan M.K., Ismaeel A.G., Al-Haddad S.A.R., Alani S. et al. A multi-layer cnn-gruskip model based on transformer for spatial-temporal traffic flow prediction. *Ain Shams Engineering Journal*. 2024. vol. 15. no. 12. p. 103045.

19. Zhang X., Chan K.W., Li H., Wang H., Qiu J., Wang G. Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model. *IEEE transactions on cybernetics*. 2020. vol. 51. no. 6. pp. 3157–3170.
20. Yang Y.-Q., Lin J., Zheng Y.-B. Short-time traffic forecasting in tourist service areas based on a cnn and gru neural network. *Applied Sciences*. 2022. vol. 12. no. 18. p. 9114.
21. Guo F., Wei Q., Wang M., Guo Z., Wallace S.W. Deep attention models with dimension-reduction and gate mechanisms for solving practical time-dependent vehicle routing problems. *Transportation research part E: logistics and transportation review*. 2023. vol. 173. p. 103095.
22. Zafar N., Haq I.U., Chughtai J.-u.-R., Shafiq O. Applying hybrid lstm-gru model based on heterogeneous data sources for traffic speed prediction in urban areas. *Sensors*. 2022. vol. 22. no. 9. p. 3348.
23. Hou M., Liu S., Zheng Q., Liu C., Zhang X., Kang C. A deep learning based communication traffic prediction approach for smart monitoring of distributed energy resources in virtual power plants. *IET Smart Grid*. 2024. vol. 7. no. 5. pp. 653–671.
24. Hung N.V., Luu D.L., Pham Q.T., Lupu C. Comparative analysis of different spacing policies for longitudinal control in vehicle platooning. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. 2024. DOI: 10.1177/09544070241273985.
25. Kool W., Van Hoof H., Welling M. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*. 2018.
26. Cheng Q., Yuangyai C., Zhan Y.-Q., Cheng C.-Y. Optimizing vehicle routing by incorporating driver preferences: A pointer network and deep q network (pndqn) approach.
27. Li K., Zhang T., Wang R. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*. 2020. vol. 51. no. 6. pp. 3103–3114.
28. Jiang W., Han H., Zhang Y., Wang J., He M., Gu W., Mu J., Cheng X. Graph neural networks for routing optimization: Challenges and opportunities. *Sustainability*. 2024. vol. 16. no. 21. p. 9239. DOI: 10.3390/su16219239.
29. Bahovska E. Graph neural networks in neighbourhood selection for a vehicle routing problem solver. *Master's thesis*. 2023.
30. Tien Z.C., Qi-lee J. Enhancing vehicle routing problem solutions through deep reinforcement learning and graph neural networks. *International Journal of Enterprise Modelling*. 2022. vol. 16. no. 3. pp. 125–135.
31. 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. p. e2. DOI: 10.4108/eetinis.v9i4.2218.
32. Inzillo V., Garompolo D., Giglio C. Enhancing smart city connectivity: A multi-metric cnn-lstm beamforming based approach to optimize dynamic source routing in 6g networks for MANETs and VANETs. *Smart Cities*. 2024. vol. 7. no. 5. pp. 3022–3054.
33. “solomon-100”. URL: <https://www.sintef.no/globalassets/project/top/vrptw/solomon/solomon-100.zip> (дата обращения: 30.05.2025).
34. “solomon-200”. URL: https://www.sintef.no/globalassets/project/top/vrptw/homberger/200/homberger_200_customer_instances.zip (дата обращения: 30.05.2025).
35. “solomon-400”. URL: https://www.sintef.no/globalassets/project/top/vrptw/homberger/400/homberger_400_customer_instances.zip (дата обращения: 30.05.2025).
36. Weisstein E.W. Distance. html, 2003, from MathWorld – A Wolfram Web Resource. Available at: <https://mathworld.wolfram.com/Distance> (accessed 30.05.2025).

37. Solomon M.M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*. 1987. vol. 35. no. 2. pp. 166–324.
38. Kim Y.H., Kim I., Kim Y.-J., Kim M., Cho J.-H., Hong M., Kang K.-H., Lim S.-H., Kim S.-J., Kim N. et al. The prediction of sagittal chin point relapse following two-jaw surgery using machine learning. *Scientific Reports*. 2023. vol. 13. no. 1. DOI: 10.1038/s41598-023-44207-2.
39. Sulistio B., Warnars H.L.H.S., Gaol F.L., Soewito B. Energy sector stock price prediction using the CNN, GRU & LSTM hybrid algorithm. *International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*. IEEE, 2023. pp. 178–182.
40. Gorishniy Y., Rubachev I., Khrulkov V., Babenko A. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*. 2021. vol. 34. pp. 18932–18943.
41. Popov S., Morozov S., Babenko A. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*. 2019.

Хунг Нгуен Вьет — Ph.D., преподаватель, международный институт подготовки кадров и сотрудничества, Восточноазиатский технологический университет. Область научных интересов: мультимедийные коммуникации, сетевая безопасность, искусственный интеллект, организация трафика в сетях нового поколения, гарантия качества сетевых услуг, экологичные сети, приложения. Число научных публикаций — 30. hungnv@eaut.edu.vn; Ки Ань, Хатинь, Вьетнам; р.т.: +84(098)911-2079.

Ту Хуонг Тринь — преподаватель, международный институт подготовки кадров и сотрудничества, Восточноазиатский технологический университет. Область научных интересов: логистика и управление цепями поставок, деловое администрирование, экономика и менеджмент. Число научных публикаций — 2. huongtt2@eaut.edu.vn; Тханьхоа, Вьетнам; р.т.: +84(090)309-4888.

Тан Нгуен — научный сотрудник, Восточноазиатский технологический университет. Область научных интересов: приложения, анализ данных. Число научных публикаций — 4. tan25102000@gmail.com; Хынгйен, Вьетнам; р.т.: +84(035)919-0216.

Доан Труонг Конг — Ph.D., преподаватель, международная школа, Ханойский университет науки. Область научных интересов: прикладной искусственный интеллект, машинное обучение и наука о данных в финансах, маркетинге, менеджменте. Число научных публикаций — 30. tcdoan@vnu.edu.vn; Ханой, Вьетнам; р.т.: +84(090)215-4359.

Нам-Хоанг Нгуен — доцент, факультет информационных технологий, Восточноазиатский технологический университет. Область научных интересов: сети мобильной связи, Интернет вещей, связь с помощью видимого света. Число научных публикаций — 47. hoannguyen@eaut.edu.vn; Ханой, Вьетнам; р.т.: +84(24)3754-9338.

A. KOKATE, T. JADHAV

A NOVEL APPROACH TO EEG ARTIFACT REMOVAL USING ADASYN AND OPTIMIZED HIERARCHICAL 1D CNN*Kokate A., Jadhav T. A Novel Approach to EEG Artifact Removal Using ADASYN and Optimized Hierarchical 1D CNN.*

Abstract. In neuroscience, neural engineering, and biomedical engineering, electroencephalography (EEG) is widely used because of its non-invasiveness, high temporal resolution, and affordability. However, noise and physiological artifacts, such as cardiac, myogenic, and ocular artifacts, frequently contaminate raw EEG data. Deep learning (DL)-based denoising techniques can reduce or eliminate these artifacts, which degrade the EEG signal. Despite these techniques, significant artifacts can still hinder the performance, making noise removal a major requirement for accurate EEG analysis. Furthermore, for strong artifact removal, an Optimized Hierarchical 1D Convolutional Neural Network (1D CNN) is introduced. For effective feature extraction, the hierarchical CNN combines max-pooling, ReLU activation, and adaptive convolutional windows. An Annealed Grasshopper Algorithm (AGA) is employed to optimize the network parameters, further improving artifact removal. To ensure comprehensive exploration and convergence toward ideal CNN settings, AGA combines the fine-tuning accuracy of Simulated Annealing (SA) with the global exploration capabilities of the Grasshopper Optimization Algorithm (GOA). By utilizing a hybrid technique, the network can more effectively eliminate artifacts from various hierarchical levels, leading to a notable improvement in signal clarity and overall accuracy. The cleaned EEG data is represented by the recovered features in the last dense layer of the Hierarchical 1D CNN, which employs a sigmoid function. Based on experimental results, the proposed method achieved a PSNR of 29.5dB, MAE of 11.32, RMSE of 0.011, and CC of 0.93, which outperforms prior works. The proposed method can improve the precision of EEG artifact removal, which is a useful addition to biomedical signal processing and neuro-engineering.

Keywords: electroencephalography (EEG), signal processing, Convolutional Neural Network (CNN), Simulated Annealing (SA), Grasshopper Optimization Algorithm (GOA).

1. Introduction. A variety of electrodes, including sticky and dry electrodes, can be used to record EEG signals. Brain-computer interfaces (BCIs) can be classified as invasive, non-invasive, or semi-invasive depending on where the electrodes are placed [1]. Since electrodes are applied to the scalp, non-invasive methods are most often utilized in medical diagnosis, research studies, and many different BCI systems [2]. Electrodes applied to the scalp typically cause a significant amount of signal artifacts, which contaminate the signal. To create a BCI system that is effective, these artifacts must be eliminated [3]. The artifacts can be detected and eliminated using a variety of techniques. These techniques ought to eliminate the artifacts while maintaining the EEG signal's original neural activity [4]. The possibility of EEG data as a diagnostic and monitoring tool for a range of medical uses has been demonstrated. These applications include the measurement of anesthesia levels neuro-feedback prior to and during surgery [5], the recognition of epilepsy [6], predicting

the onset of epileptic seizures [7], neuro-feedback applications for patients with autism [8], and neuro-rehabilitation [9]. However, the EEG has been characterized by several inherent problems, such as the need to eliminate additive noises, or EEG artifacts, which can be produced by a variety of noise sources, including muscular movements or interference from power lines [10]. The primary purpose of the EEG artifact removal techniques is to remove artifacts from the EEG data. The full utilization of EEG data for clinical and industrial applications is made possible by the effectiveness of EEG artifact removal techniques [11].

There are several kinds of issues with the EEG artifact removal techniques. The nonlinearities of the noise being added to the EEG signal or the complexity of the procedures may be the cause of these difficulties [12]. The non-stationary and non-linear nature of the EEG signal makes it challenging to detect artifacts without sacrificing neural information. Processing the signal becomes challenging due to artifacts that can affect its spectral, temporal, and occasionally spatial domains [13]. In this instance, artifacts cannot be entirely eliminated during pre-processing by simple filtering. There is still no technique that can find and eliminate every kind of artifact, despite the development of numerous hybrid approaches [14]. EEG artifacts have been eliminated by recent researchers using various Time-Frequency Representation (TFR)-based signal decomposition techniques. To extract the fine-scale fluctuations in EEG signals, several TFR techniques are used, such as Wavelet Transform (WT), Short Time Fourier Transform (STFT), etc. [15]. EMG artifacts are removed using a variety of techniques, the most widely used being signal decomposition approaches and Blind Source Separation (BSS) techniques like Canonical Correlation Analysis (CCA) and ICA. This experimentation yields validated computation time and accuracy results after the artifacts are effectively removed using both BSS methods [16].

Currently, instead of exploring ways to modify the conventional methods, researchers are more focused on combining different traditional algorithms that already exist to create hybrid techniques. The process of eliminating artifacts has been made more accurate, automated, and efficient by combining the algorithm's beneficial features [17]. As of yet, standard norms or optimal methods for artifact elimination have not been established, although neurologists may find this to be crucial for a successful clinical diagnosis [18]. Despite its superior performance and reasonable results, this model's dependence on the threshold function and wavelet form causes data loss in EEG signals [19]. Numerous obstacles must be addressed by existing models. As a result, DL is employed to address problems with traditional approaches [20].

Potential techniques for automatically extracting complex data characteristics at high degrees of abstraction have been made possible by recent advances in DL methodology [21]. Additionally, it offers several methods for effectively eliminating ocular artifacts from EEG signals. The benefits of utilizing DL techniques include strong generalization ability, time savings, and the elimination of the need for additional EOG reference signals, among others [22]. When it comes to identifying and reducing EEG signal artifacts, the majority of DL models offer high clearance. The decoding and classification of EEG signals, which are typically associated with low signal-to-noise ratios (SNRs) and high data dimensionality, have been the focus of DL frameworks in recent times due to the growing availability of large EEG datasets. Hence, there is a need to develop a novel network to overcome the aforementioned issues in the removal of artifacts from EEG signal.

The key contributions of this work are as follows:

- Introduces a novel DL-based architecture that addresses EEG artifact removal, enhances signal clarity, and balances class distributions in difficult feature space areas by merging a Hierarchical 1D CNN and the ADASYN.
- To enhance the model's capacity to precisely extract features and eliminate artifacts from EEG data by introducing an optimized CNN with adaptive windows, max-pooling ReLU activation, and a sigmoid classifier.
- To optimize CNN parameters for effective artifact removal and prevent overfitting, this method leverages the global search ability of GOA and the fine-tuning precision of SA.

The remaining work is structured as follows: Section 2 discusses the related research from the previous work, Section 3 explains the proposed approach, Section 4 reveals the results of the proposed method and discusses previous work, and Section 5 summarizes the article.

2. Literature Survey. In paper [23] the authors proposed an intelligent model for artifact removal in EEG signals, comprising training and testing phases. The model utilized an improved 1D-CNN, with parameters fine-tuned using a hybrid optimization algorithm, SM-EFO, which combines Spider Monkey Optimization (SMO) and Electric Fish Optimization (EFO). However, the model faced potential overfitting and increased computational complexity due to the integration of multiple optimization algorithms.

Study [24] developed a deep 1D CNN for automatic identification of abnormal and normal EEG patterns. The model's integration with

optimization mechanisms added complexity and demanded high computational resources, especially during long-term data analysis.

Paper [25] implemented a multistage adaptive cascaded noise canceller to remove artifacts such as baseline wander, motion, muscle, and power line interference from ECG signals. Despite its performance, relying on a single algorithm to update filter weights across all stages limited adaptability to different artifact types.

In paper [26] the authors introduced a deep learning-based model to detect and remove ocular artifacts using Pisarenko harmonic decomposition, DWT, PCA, and ICA for feature extraction. An optimized DCN, fine-tuned via DS-EFO, was employed for artifact removal. Nonetheless, the model showed a slight drop in accuracy and needed further improvements for effectively detecting ocular artifacts.

In paper [27] the author addressed ECG-induced artifacts in EEG using the RVFLN method and RLS algorithm. Although adaptive filtering was achieved, the approach was susceptible to failure when the artifact frequency was unknown or difficult to determine.

Paper [28] developed a DNN-based model with convolutional layers for artifact removal from long-term EEG recordings sourced from the EPILEPSIAE database. However, the method was mostly applicable to data acquired at similar sampling rates, limiting its broader applicability.

Study [29] combined an LSTM network with a kNN classifier to detect and remove eye-blink and muscle artifacts from EEG signals. Artifact detection was based on features such as peak-to-peak amplitude and variance. However, expert-averaged evaluations were recommended to improve accuracy and system training.

Paper [23] introduced a model named “AnEEG” for artifact removal using deep learning, outperforming traditional wavelet methods. Yet, it required manual identification of artifact frequency patterns, which reduced adaptability to unknown artifact conditions and limited generalizability.

Study [30] presented LSTEEG, an LSTM-based autoencoder for artifact detection and correction in EEG signals. The model effectively captured non-linear dependencies and enhanced downstream EEG processing. Nonetheless, its reliance on previously seen artifact patterns limited its performance on novel artifact types.

In paper [31] the authors proposed DWINet, which utilized the image dehazing capabilities of DRHNet by treating EEG denoising as an image-processing problem. Although effective in general denoising, DWINet underperformed in handling muscle artifacts due to the model's visual-domain foundation.

Overall, earlier EEG artifact removal techniques faced challenges such as overfitting from hybrid optimization, high computational demands, limited adaptability to various artifact types, sensitivity to artifact frequency estimation, and constraints tied to acquisition setups. These limitations motivated the development of a novel network to address these issues and enhance artifact removal performance in EEG analysis.

3. Proposed Method. This study integrates Adaptive Synthetic Sampling (ADASYN) to address class imbalance by generating synthetic samples in complex feature spaces, ensuring balanced data representation. It incorporates an Optimized Hierarchical 1D Convolutional Neural Network (CNN), which uses adaptive convolutional windows, max-pooling, and ReLU activation to extract relevant features. The final dense layer with sigmoid activation delivers the cleaned EEG output. Furthermore, the Annealed Grasshopper Algorithm (AGA) is employed to fine-tune CNN parameters by combining the global search capabilities of the Grasshopper Optimization Algorithm (GOA) with the precise local optimization of Simulated Annealing (SA). AGA dynamically tunes the kernel sizes, strides, and learning parameters of the CNN during training, ensuring that the network remains effective across various artifact conditions. This hybrid optimization strategy dynamically adjusts the CNN's convolutional window sizes, pooling parameters, and learning rates based on the EEG signal's noise characteristics, allowing for effective generalization across different artifact types. The proposed method's block diagram is presented in Figure 1.

Figure 1 illustrates the proposed signal denoising and optimization framework. The raw input signal is first processed using Adaptive Synthetic Sampling (ADASYN) to address class imbalance by generating synthetic minority samples. The balanced data is then passed into the Optimized Hierarchical 1D CNN, which is composed of convolutional, ReLU, max-pooling, flatten, dense, and sigmoid layers to extract relevant features. The CNN parameters are tuned using the Annealed Grasshopper Algorithm (AGA), which optimizes hyperparameters such as kernel size, stride, pooling configuration, and learning rate, with updates guided by binary cross-entropy loss until the stopping criteria are satisfied. The resulting optimized CNN parameters are then applied in the final training phase.

The cleaned, denoised EEG waveform refers to the reconstructed signal obtained after processing by the optimized CNN. In this stage, unwanted artifacts such as ocular (EOG), muscular (EMG), and cardiac (ECG) noise are removed, while the essential brain activity patterns are preserved. This denoised output provides a more accurate representation of neural activity, making it suitable for further clinical or research analysis.

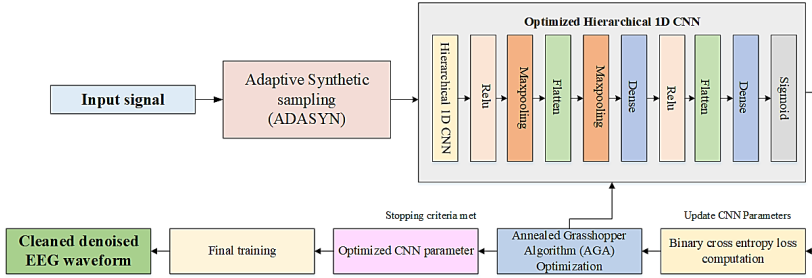


Fig. 1. Schematic block for the proposed approach

3.1. Adaptive Synthetic Sampling Approach (ADASYN). The majority of datasets used for classification are imbalanced. One group (the minority class) is underrepresented in the dataset, whereas the majority class has more data in the other set. Algorithms for ML cannot function well in these situations. Data sampling techniques are used to classify unbalanced data sets to improve the efficiency of ML classification algorithms. The SMOTE method has been enhanced with the ADASYN method. It does the following tasks: It operates on two sets of data. The data in the minority class roughly corresponds to the data in the majority class for both the minority and majority classes [30].

Input: The training data set D_{tr} contains n samples $\{a_i, b_i\}$, $i = 1, \dots, n$, where $b_i \in B = \{1, -1\}$ indicates label of class identity connected to a_i , and a_i indicates within the n -dimensional space of features, A . Define Maj_c , and Min_c as the number of examples from the minority class and the number from the majority class, respectively. Consequently, $Maj_c > Min_c$ and $Maj_c + Min_c = M$.

Determine the degree of the class imbalance using (1):

$$D_{CI} = Maj_c / Min_c, \quad (1)$$

where $D_{CI} \in (0, 1]$.

When $D_{CI} < D_{CIth}$, which is a predetermined threshold for the highest amount of class imbalance ratio that can be tolerated, then, determine the number of artificial data instances required for the minority class using (2):

$$G = (Min_c - Maj_c) \times \gamma, \quad (2)$$

where γ indicates the factor specifies the intended balance level following the production of the artificial data, which has a range of $[0, 1]$. The generalization procedure results in the production of a fully balanced data set if $\gamma = 1$. Determine the K nearest neighbors for each example $a_i \in \text{minority class}$ using the Euclidean distance in n -dimensional space. Then, compute the ratio r_i which is defined as (3):

$$r_i = \Delta i / K, i = 1, \dots, Maj_c, \quad (3)$$

where $r_i \in [0, 1]$ because Δi indicates the number of examples in the K nearest neighbors of a_i that are members of the majority class;

To make $\hat{r}_i$ a density distribution ($\sum_{i=1}^{Maj_c} r_i = 1$), normalize r_i by $\hat{r}_i = r_i / \sum_{i=1}^{Maj_c} r_i$.

For every minority example a_i , determine how many synthetic data examples must be created using (4):

$$g_i = \hat{r}_i \times G, \quad (4)$$

where G indicates the total amount of instances of artificial information required to be created, as per (2), for the minority class.

Create g_i synthetic data examples for each minority class data example a_i using the procedures listed below:

From 1 to g_i , complete the loop:

(i) For data a_i , randomly select one minority data example (a_{zi}) from the K nearest neighbors.

(ii) Produce the example of synthetic data using (5):

(iii)

$$s_i = a_i + (a_{zi} - a_i) \times \lambda. \quad (5)$$

In n -dimensional spaces, the difference vector is represented by $(a_{zi} - a_i)$, and λ is a random number with a range of $[0, 1]$. Close the Loop.

The fundamental principle of the ADASYN algorithm is that the quantity of synthetic samples needed for each minority data should be automatically calculated. Following ADASYN, the dataset will display a distribution of data that is balanced based on the β coefficient's desired balance level. Additionally, the learning algorithm will be forced to focus

on the examples that can be difficult to learn. In contrast to the SMOTE algorithm, which generates an equivalent quantity of synthetic samples for each instance of minority data, this is a significant departure. The preprocessed signal is fed to the CNN and will be discussed below.

3.2. Optimized Hierarchical 1D CNN. Over the past few decades, 1D-CNNs have become increasingly popular within the DL field. It has been used for raw continuous signals in a variety of contexts by effectively removing various signals' artifacts. Several features make up the 1D-CNN architecture, including enhanced spatiotemporal feature mining structure, automatic feature learning to achieve adaptive design, and faster classification accuracy [31]. Because of the condensed and straightforward design of 1D-CNNs, they also exhibit feasible efficacy concerning affordable hardware and instantaneous software. The only 1D convolutions that can be performed with this model are scalar additions and multiplications. EEG signal denoising is better with 1D-CNN, particularly for prolonged sections. It generally uses an end-to-end design to eliminate EEG signal artifacts. Ultimately, the noisy signal is rebuilt to produce the network output. Since the time sequences of EEG signals are independent with only one dimension, a 1D-CNN is employed to remove artifacts via the 1D convolution layer. In the 1D-CNN architecture, a fixed-size overlapping window is used to separate the signals from the EEG into sub-signals. It is made up of fully connected layers, maximum pooling layers, and various convolution layers. The convolution layer convolves the output of the feature vector using the preceding layer's convolution kernel. The output feature vector is created using the non-linear activation function.

The input signal sequence is represented as $\vec{S}_{S_i}$, where $i = 1, 2, \dots, m_i$, and the filter is represented as F_i with $i = 1, 2, \dots, n$. This means that the length of the filter n must be less than the length of the signal sequence m_i . Partial convolution is used to perform the filter depending on the preceding layer's input features. (6) formulates the 1d-CNN's convolved output a_i .

$$a_i = \sum_{g=1}^n F_g \times \hat{S}_{S_i-g+1}. \quad (6)$$

In this instance, the local connection network is formed by correlating every neuron in the d^{th} layer with neurons in the $(d - 1)^{th}$ layer of the local window. The non-linear mapping is carried out by the activation function $af(\hat{S}_S)$ in the convolution layer. (7) describes the function that an activation, a modified linear unit, uses in this 1D-CNN model to increase convergence speed.

$$af(\mathcal{S}_s) = \max(0, \mathcal{S}_s). \quad (7)$$

Additionally, (8) derives the input of the q^{th} neurons in the d^{th} layer.

$$C_q^d = af(\sum_{r=1}^n F_r^d \times C_{q-r+1}^{d-1} + Op^q) = af(\vec{F}^d \times \vec{C}_{(q-n+1:q)}^{d-1} + Op^q). \quad (8)$$

In the aforementioned formula, the m^{th} dimension filter is denoted by $\vec{F}^d \in \mathcal{R}^m$, which is the same for each neuron in the convolution layer. The offset parameters are denoted as Op^q , where $q = 1, 2, \dots, n_i$. Due to many benefits, including simplified array activities, easier training, and easier execution all contribute to decreased computational complexity with a small number of hidden layers, 1D-CNNs operate effectively in 1D EEG signals. The model has demonstrated improved performance in extracting clean signals from noisy inputs.

The 1D-CNN architecture offers distinct features for removing artifacts from EEG signals, but it also presents several difficulties, including complexity, being prone to error, and a lack of noise elimination features, and the unique characteristics of EEG signals, such as diversity, time variation, uncertainty, and nonlinearities, which make them difficult to process linearly. In the same way, it is essential to use non-linear denoising for EEG signals. Furthermore, because of the gradient explosion issue and degradation phenomenon, training the deep network is essential to getting the intended outcome. In addition, since EEG signals are typically lengthy and intricate 1D signals, an effective 1D CNN must be created in this model to extract the more intricate features of the artifacts removal model from EEG signals that are non-linear. By combining the benefits of CNN with the non-linear properties of EEG signals that change over time, a new and improved 1D-CNN utilizing the AGA is proposed as a solution to these problems. This developed 1D-CNN immediately learns its biased and non-linear deep features from the EEG signals that have various artifacts. After that, the learned features are used to distinguish between them, and reconstruction is carried out to produce clean EEG signals. Figure 2 depicts the Optimized Hierarchical 1D CNN architecture given below.

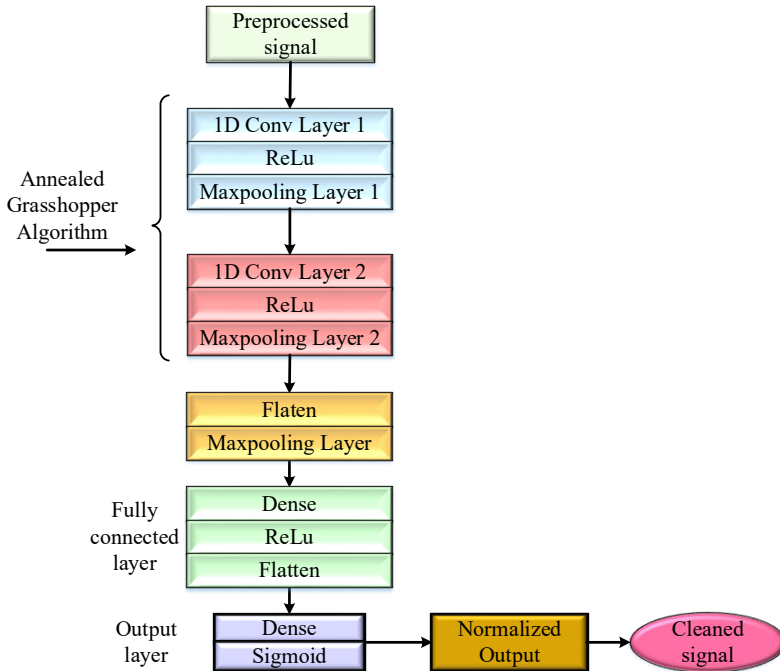


Fig. 2. Optimized Hierarchical 1D CNN architecture

The convolutional layer is the first layer in a CNN. To extract the features from the input feature map, it makes use of the receptive fieldset. A set of filters is used by the two-dimensional convolutional layer to process the sources in a narrow area with nearby interactions. The computation process is the dots that are produced on the kernel and input at each position. Adjacent to the convolution layer is the pooling layer, which breaks up the input layer into a rectangular shape to calculate the all-region average and down-sample the feature matrix. Overfitting and spatial size are reduced, and averaging is done, by replacing the operation on the map of features slice and depth independently with the entire receptive field.

The last layer in the network is the fully connected layer. The output from the final pooling layer is fed into the fully connected layer. To acquire the clean signals, the upgraded 1D-CNN is fed the 400×1 window containing the noisy EEG signals. (9) describes this procedure. Let's assume that $\widehat{CS}_t$ is made up of pure EEG signals ($\widehat{S}_t$) and artifacts ($n\widehat{S}_t$).

$$\widehat{CS}_t = \widehat{S}_t + n\widehat{S}_t. \quad (9)$$

The clean and noisy EEG signals are represented as $\widehat{S}_{S_t}$ and $n\widehat{S}_{S_t}$, respectively, in this case, with both types of signals included in $\mathcal{C}\widehat{S}_{S_t}$. To achieve artifacts removed signal $\widehat{S}_{S_t}$, the 1D-CNN trains the network parameters and creates a more complex non-linear function $af(\Theta)$, which minimizes the error function. To reconstruct EEG signals, the created model is primarily concerned with mapping function learning $af(a)$.

3.2.1. Annealed Grasshopper Algorithm. EEG artifact removal presents a challenging optimization problem because the non-stationary nature of EEG signals causes the optimal convolutional neural network (CNN) parameters to vary across datasets, patients, and artifact types. Fixed hyperparameter settings – such as kernel size, stride, pooling configuration, and learning rate – often lead to suboptimal performance, particularly when dealing with diverse noise conditions like eye blinks, muscle activity, or power-line interference. Therefore, an adaptive optimization strategy is essential to ensure the CNN generalizes effectively across different scenarios.

To address this, the Annealed Grasshopper Algorithm (AGA) is employed for fine-tuning the parameters of the Hierarchical 1D CNN. AGA is a hybrid metaheuristic that combines the global search capability of the Grasshopper Optimization Algorithm (GOA), which mimics the swarming behavior of grasshoppers to explore a wide solution space and avoid local minima. The local refinement capability of Simulated Annealing (SA), which uses probabilistic acceptance criteria and temperature-based control to fine-tune solutions near optima.

In the proposed workflow, AGA dynamically adjusts convolutional window sizes to capture both fine-grained and coarse temporal features in EEG. Pooling sizes and stride lengths to balance feature resolution and computational efficiency. Learning rate for stable yet responsive model updates. Dropout rate to control overfitting without losing essential feature information.

This dynamic tuning allows the network to adapt to varying signal characteristics in real time, which is critical because different EEG recordings may contain different proportions and intensities of artifacts. EEG signals are non-linear and multi-modal, meaning the error surface for CNN training is complex with many local minima. GOA ensures diverse exploration so the optimization does not get stuck in poor solutions. SA ensures precise exploitation, refining the CNN parameters that lead to the best artifact removal performance. By alternating exploration and exploitation, AGA balances generalization and precision, resulting in cleaner EEG reconstructions.

A. Grasshopper Optimization Algorithm. One new algorithm, GOA, imitates the grasshoppers' swarming behavior. Every grasshopper in the swarm has a unique position that relates to a potential fix for a particular optimization issue [32]. A_i represents the position of the i -th grasshopper where ($i=1,2,3,\dots,n$) in (10).

$$A_i = S_i + G_i + W_i. \quad (10)$$

S_i represents social interaction, G_i represents the force of gravity acting on a i -th grasshopper, and W_i represents wind advection. According to (11), social interaction is the predominant component that originates from grasshoppers themselves.

$$S_i = \sum_{j=1, j \neq i}^N s(d_{ij}) \cdot \hat{d}_{ij}, \quad (11)$$

where d_{ij} is the distance between the i -th and j -th grasshopper, $\hat{d}_{ij}$ is a unit vector between the i -th and j -th grasshopper.

The gravitational constant, gc , and the unity vector ($\hat{e}_{gc}$), which points toward the center of the earth, make up the two components of the G_i component. (12) defines mathematics.

$$G_i = -gc \cdot \hat{e}_{gc}. \quad (12)$$

This is the calculation used to get the wind advection W_i using (13)

$$W_i = df_c \cdot \hat{e}_w, \quad (13)$$

where $\hat{e}_w$ is a unity vector in the wind direction and df_c is a constant drift. (10) can be written as follows using components as (14):

$$A_i = \sum_{j=1, j \neq i}^N s(d_{ij}) \cdot \hat{d}_{ij} - gc \cdot \hat{e}_{gc} + df_c \cdot \hat{e}_w. \quad (14)$$

In a stochastic algorithm, finding a medium base between exploration and exploitation aids in locating the global optimum. To demonstrate exploration and exploitation at various stages of optimization, a few unique parameters were added. (15)'s mathematical model becomes:

$$A_i^d = E \cdot \left(\sum_{j \neq i}^N \left[\frac{E(u_d - l_d)}{2} \cdot s(d_{ij}) \cdot \hat{d}_{ij} \right] \right) + tar_d, \quad (15)$$

where the lowering factor "E" is employed twice in (6) to regulate forces between grasshoppers and changed with (16). Here, GF is disregarded, considering no gravitational force and wind direction remains towards a target. The inner "E" minimizes the repulsion/attraction forces between grasshoppers proportional to the number of iterations, while the outer "E" preserves a compromise between exploration and exploitation, where the target's d-th dimension value is represented by tar_d , the lower bound in the d-th dimensions is represented by l_d , and the upper bound is represented by u_d (the most beneficial solution discovered consequently well).

$$E = E_{max} - \frac{CI \cdot (E_{max} - E_{min})}{I}, \quad (16)$$

where I denotes the maximum number of iterations, CI denotes the current iteration, and $E_{min} = 0.00001$ denotes the minimum value.

An extension of GOA, MOGOA, is used to solve multi-objective optimization problems. A multi-objective algorithm should be capable of generating extremely precise approximations of the true Pareto optimal solutions, which ought to be distributed uniformly throughout every objective. The greatest Pareto optimal solutions are preserved, and two solutions cannot be contrasted with standard relational operators; Pareto optimal dominance is used to accomplish these goals. Selecting the target was the primary obstacle in the creation of MOGOA. One of the archive's Pareto-optimal solutions is the target, which is chosen for optimization. (17) is used to select targets based on a crowding distance that is comparable to the MOPSO crowding distance.

$$P_i = \frac{1}{n_{ni}}, \quad (17)$$

n_{ni} is the number of solutions nearby of the i th solution, and P_i is the probability of selecting the target from the archive. Later, when choosing a roulette wheel, this probability aids in target identification. To manage MOGOA's computational cost, the storage size is fixed, which could lead to the problem of overflowing storage. To solve this problem, solutions in the more populated areas of the store are once more eliminated using the

inverse of P_i and a roulette wheel. In this way, the storage space is updated regularly.

B. Simulated Annealing. Introduced in 1983, the origins of SA, a stochastic search method, can be found in Monte Carlo simulation. It can solve difficult combinatorial optimization problems [33]. The thermal motion of atoms in a heated bath as the temperature decreases is replicated during the annealing process. Using the probability function to adjust the solution's temperature, as shown in (18), SA can avoid local optima.

$$P(\Delta E) = e^{-\frac{\Delta E n}{TK_B}}, \quad (18)$$

where T is the present temperature, En is the atomic energy, and K_B is Boltzmann's constant. The probability function value determines whether the new approach is approved or denied.

Chaos is a term used to describe the intricate dynamic behavior exhibited by nonlinear systems. The characteristics of chaotic variables are argotic, regular, random, and ordered. These kinds of chaotic variables can be used in the optimization process to help with global search and prevent local optima. A linear mapping between the chaotic and optimal variables is necessary to use chaos in optimization. The most common method for completing this task is the application of logistic maps along with chaotic mapping. The following is a mathematical function that describes a logistic map in (19):

$$Z_{i+1} = \mu \times Z_i \times (1 - Z_i), \quad (19)$$

where Z_i is the value of Z in iteration i , Z_{i+1} is the new value of Z , and $3.57 < \mu \leq 4$, with $\mu = 4$ yielding the best results. The initial value of Z is set to $rand()$ in the interval $[0, 1]$.

C. Annealed Grasshopper Algorithm (AGA). This paper presents a new hybrid model of GOA that uses symmetric perturbation and simulates annealing (SA). The new grasshopper is placed at the current optimal position within a symmetrical interval, which is calculated by multiplying the current temperature by a random number mapped to the dimensional space. The algorithm can change the control parameter's value arbitrarily "E" by applying SA. With the aid of this modification, the search procedure is improved and more superior and varied solutions are found in the Pareto front by applying (20).

$$E_{new} = E_{old} * (1 + N_s) * e^{-N_s * \frac{1}{N_{GSM}}}, \quad (20)$$

where E_{old} is the value from the previous iteration and E_{new} is the new perturbation "E"; E_{old} is unchanged but modified in the initial iteration in each subsequent iteration. N_{GSM} is the number of grasshoppers in the swarm, and N_s is the number of steps in SA. The temperature is changed during the annealing process by using (21).

$$T_A = T_A * \beta. \quad (21)$$

The cooling coefficient, $\beta \in (0, 1)$, in (21) lowers the temperature with each iteration. After using SA to alter the inertia weight value, β was set to 0.95. A new value for "E" is embraced if population fitness increases; otherwise, the Gaussian probability function, as indicated in (22), is used to calculate probability.

$$G_{PF}(t) = \min \left(1, e^{-\left(\frac{F_{new} - F_{old}}{K_B T_A} \right)} \right), \quad (22)$$

where T_A is the annealing temperature, K_B is Boltzmann's constant, F_{old} is the fitness from the previous iteration, and F_{new} is the fitness after obtaining a new value of E using (20).

(23) modifies "E" using $G_{PF}(t)$, and the subsequent iteration begins.

$$E_{new} = E_{old} * G_{PF}(t). \quad (23)$$

The MOGOA algorithm employs the updated values of 'E' acquired via the SA process to modify the positions of grasshoppers, thereby expediting the algorithm's convergence. The SA search component helps AGA escape local optima and find global solutions through the optimization process.

This algorithm uses chaos to create a variation on AGA. (19) provides a logistic map that adjusts the cooling coefficient α rather than a constant value. (21) will thus become (24).

$$\beta_{i+1} = \mu \times \beta_i \times (1 - \beta_i), \quad (24)$$

$$T_A = T_A \times \beta_{i+1}. \quad (25)$$

Unlike the original logistic map, which used $rand()$, the modified logistic map sets the initial value of β_i to 0.95. The new value T_A produced

by applying (25) is then used in the SA procedure. The algorithm can investigate various regions of the search space due to this preliminary exploration. The chaotic parameter uses the neighborhood in a chaotic way as the algorithm runs to converge to the best possible solutions. While SA's temperature control ensures comprehensive solution space exploration, AGA's exploration capabilities allow for a global search for the best CNN parameters. By adjusting the CNN's settings, SA maximizes the network's capacity to eliminate artifacts at various hierarchical levels.

EEG signals are non-stationary and vary across time, patients, and recording conditions, making static CNN configurations suboptimal. Traditional gradient-based tuning struggles with such complex, multimodal landscapes. Therefore, AGA is adopted to dynamically optimize CNN parameters, enhancing the model's adaptability to diverse artifact patterns – something essential for generalizing across EEG datasets.

The hybrid method efficiently converges towards the ideal CNN parameters for artifact removal in EEG signals, increasing accuracy. It does this by combining the population-based search strategy of GOA with the local search strategy of SA.

4. Result and Discussion. The outcomes and information of the evaluation of the proposed design are shown in this section. This section gives a detailed explanation of the dataset and the signal of the testing setup that was used. Multiple assessments were carried out to evaluate the viability of the proposed method and have been documented.

4.1. System Configuration. Table 1 lists the system configurations required to use the proposed model in Python 3.9. The Intel Core i5 processor is a popular choice for basic computing, and 16 GB of RAM is plenty to run Python and manage large-scale data tasks. Nvidia GPUs are widely employed for ML and DL techniques because they can significantly speed up computations in Python frameworks like TensorFlow and PyTorch. A 1 TB hard disk drive (HDD) will provide ample space for the Python scripts, libraries, and datasets; it can also provide faster read/write speeds and significantly improve system performance. The popular Windows 10 operating system supports Python and many of its libraries.

Table 1. System Configuration used in implementation

Processor	:	Intel Core i5, V generation
RAM	:	16 GB
Graphics	:	Nvidia
HDD	:	1 TB
OS	:	Windows 10
Tool	:	Python 3.9

4.2. Dataset description. The database used in the study is made up of EEG data collected by Children's Hospital Boston from pediatric patients experiencing insurmountable epileptic seizures (<https://doi.org/10.13026/C2K01R>). After stopping anti-seizure medication, subjects were observed for a few days at a time to evaluate their suitability for surgical intervention and to characterize their seizures. 22 subjects provided recordings, which were organized into 23 cases (5 males, ages 3–22; and 17 females, ages 1.5–19). Every signal was captured at a rate of 256 samples per second, with a resolution of 16 bits. There are typically 23 EEG signals in files (sometimes 24 or 26). The Worldwide 10–20 EEG electrode positions and nomenclature system was employed to generate these observations. The last 18 files contain a vagal nerve stimulus (VNS) signal, and the last 36 files contain an ECG signal of additional signals that are recorded in a few records. Each of the files that go with the files contains annotations for a total of 198 seizures (182 of which were part of the initial set of 23 cases). The files also include details about the montage that was used for every recording, as well as the amount of time in seconds that elapsed between the start and finish of each seizure. The dataset split of Training and testing as 70% and 30% with a batch size of 64, a learning rate of 0.001, and the model uses the loss as binary entropy.

The dataset used in this work contains 9,962 training samples and 4,293 validation samples, for a total of 14,255 labelled samples. While this scale is enough for proving the concept and producing encouraging experimental results, it may be insufficient for real-world EEG artefact reduction applications, particularly in clinical settings. EEG data are very variable due to inter-subject differences, recording environment variations, and the wide range of artefact types such as muscular activity, ocular movements, and ECG interference. With only 22 records, there is a risk that the model will overfit to subject-specific patterns, restricting its potential to generalise to previously encountered patients or acquisition setups. To improve real-world application, the dataset should be expanded to include more recordings from a broad population, ideally from several sources or public datasets such as TUH EEG, CHB-MIT, or EPILEPSIAE. Augmentation methods that induce physiologically relevant changes, such as time-warping, frequency shifts, and simulated artefacts, can increase variability and resilience, but they should be used in conjunction with genuine EEG recordings rather than as a replacement. Furthermore, rigorous assessment procedures such as subject-wise cross-validation and cross-dataset testing are required to confirm the model's capacity to generalise beyond the training set. Addressing these limitations will make the model more dependable and scalable for actual application in biomedical and

clinical EEG processing activities. Figure 3 depicts the EEG input Signal is given below.

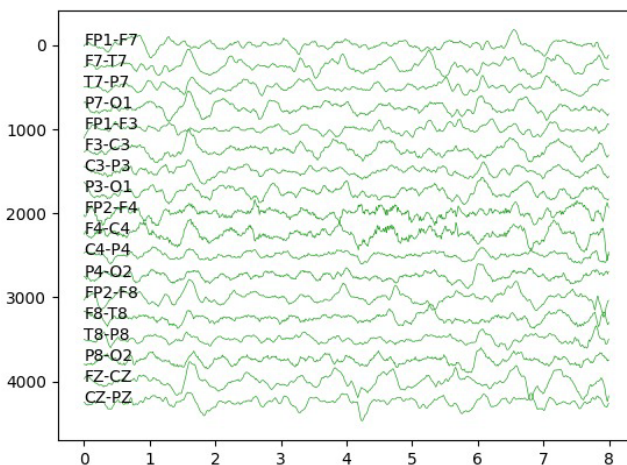


Fig. 3. EEG Input signal

4.3. Result obtained from the proposed method. Figure 4 depicts the optimized signal (denoised EEG waveform) from an Original EEG signal. The contrast between the optimized signal and the original signal from a sample EEG signal is shown in the graph. Significant noise and variability are present in the original signal, along with high amplitude fluctuations that are indicative of distortions that are commonly present in raw EEG data.

Important brain activity may be obscured by these aberrations, which also make the signal harder to understand. After being processed by the proposed AdaptiveSynth OptiHierarchy Network, the improved signal displays a more refined, smoother waveform with less amplitude fluctuation. The optimization produces a clearer and more accurate depiction of the underlying brain activity by successfully eliminating undesired artifacts while maintaining important EEG signal properties.

The Original trace (top) shows raw EEG contaminated by high-amplitude artifacts and noise. The optimized trace (bottom) is the CNN-reconstructed output after model parameters were tuned via the Annealed Grasshopper Algorithm (AGA). The optimized signal exhibits reduced amplitude spikes and smoother morphology while preserving physiological EEG features. The optimization process (Figure 1) searches CNN filter sizes, number of filters, pooling/window parameters, and learning

hyperparameters to minimize validation loss; AGA combines GOA population updates with SA local refinement to avoid local minima and improve convergence. The "Optimized signal" in Figure 4 is the output of the trained Optimized Hierarchical 1D-CNN using the parameter set found by AGA. Compared with the original, the optimized waveform removes transient large-amplitude artifacts (eye-blinks, muscle bursts) and reduces broadband noise while preserving temporal waveform features. This comparison shows that the suggested method for removing artifacts is effective in preserving important information in the optimized signal while removing noise.

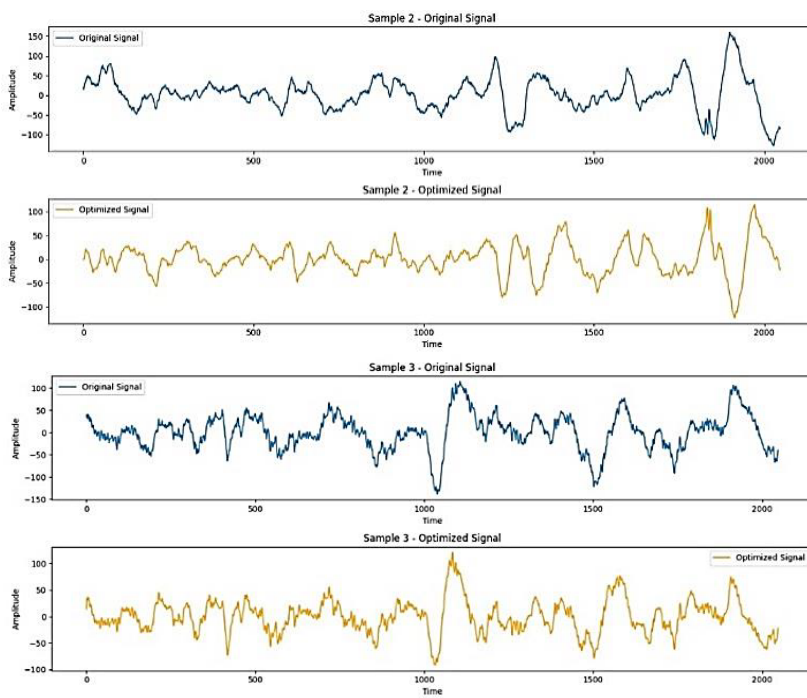


Fig. 4. Optimized signal from an Original EEG signal

An EEG signal's artifacts are gradually removed using the proposed approach, as seen in Figure 5(a). Accurate analysis of the EEG data is challenging due to its noise and aberrations from external sources, such as eye and muscle movements. Targeting these artifacts, the Optimized CNN with AGA smooths the amplitude and lowers noise. Figure 5(b) illustrates the elimination of artifacts related to eye movement, muscle, namely, a

decrease in noise. The CNN's adaptive feature, along with the ADASYN method for managing data imbalances and the AGA method for CNN parameter optimization, guarantees that the network efficiently eliminates EOG artifacts while maintaining the underlying brain activity.

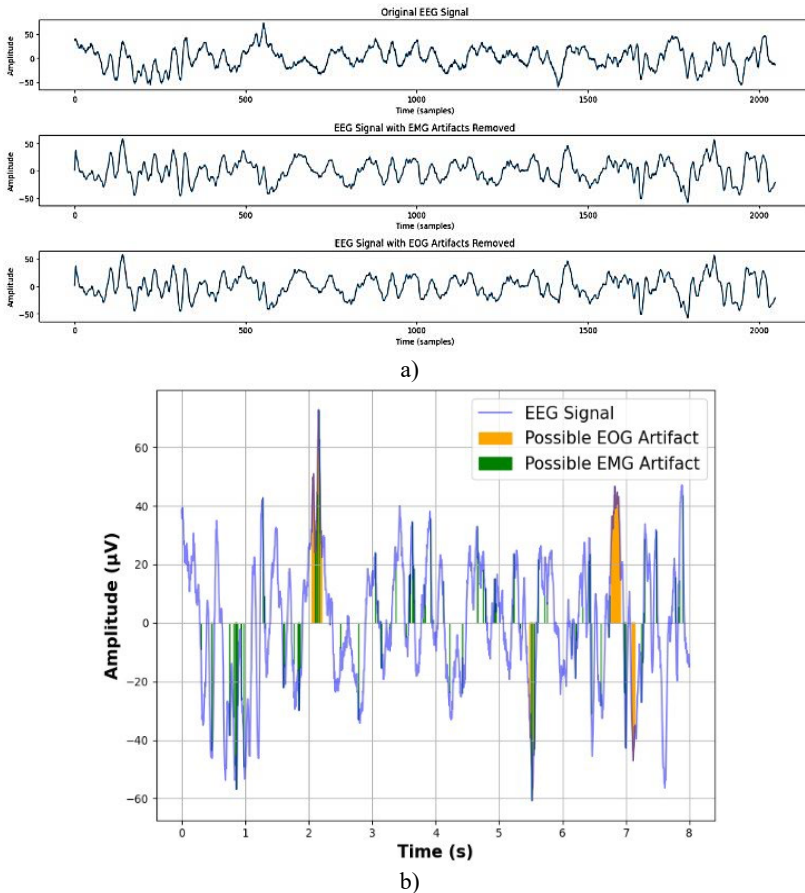


Fig. 5. a) removal of artifacts from an EEG signal; b) removal of artifacts

4.4. Performance Evaluation. Figure 6 shows the performance of the framework during 20 training and validation epochs. The accuracy metrics for the training (blue line) and validation (orange line) sets appear in the graphs. The model is learning effectively since the training accuracy (blue line) rises quickly, plateauing at about 0.989 by the fifth epoch. In the

early epochs, the validation accuracy (orange line) likewise grows dramatically; by the fifth epoch, it has reached around 0.93 before leveling off. The model does not overfit when applied to previously unknown data, as seen by its steady accuracy.

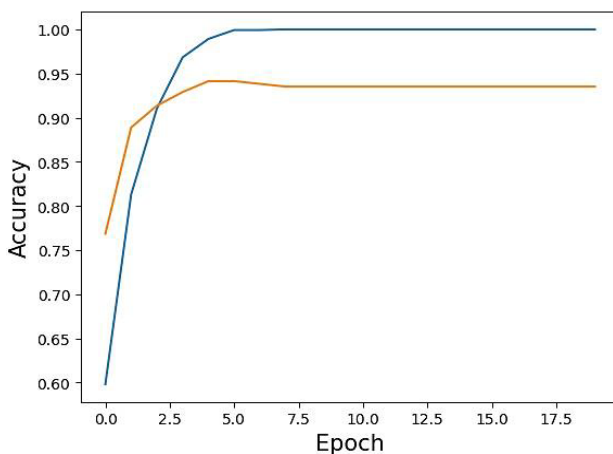


Fig. 6. Training and Validation Accuracy

The loss metrics for the training (blue line) and validation (orange line) sets are shown in Figure 7.

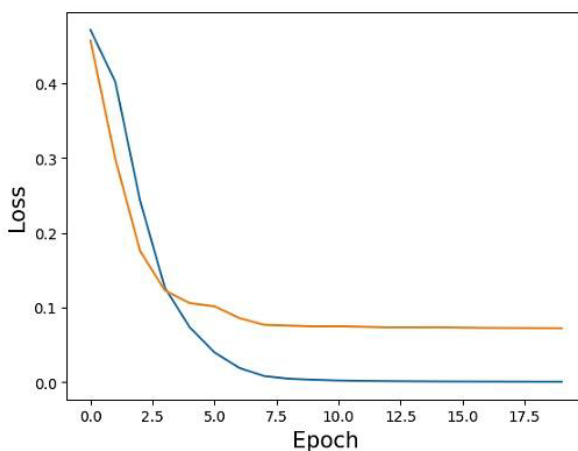


Fig. 7. Training and Validation Loss

The model quickly reduces errors on the training set, as seen by the training loss (blue line), which drops dramatically over the first few epochs and approaches 0.01 by the fifth epoch. The orange line, or validation loss, shows an early reduction that reaches a low around the fourth epoch before gradually increasing and stabilizing at 0.1. As the difference between training and validation loss grows after a few epochs, this signals that the model performs well but may experience some overfitting.

5-Fold Cross-Validation Results. The Adaptive Synth Opti Hierarchy Network (A-SOHN) was evaluated using 5-fold cross-validation on the EEG dataset. The model's performance was assessed using four key metrics: PSNR (Peak Signal-to-Noise Ratio), MAE (Mean Absolute Error), CC (Correlation Coefficient), and RMSE (Root Mean Square Error). Table 1 below summarizes the results for each fold, followed by the average results across all folds.

Table 1. 5-Fold Cross-Validation Results

Fold	PSNR	MAE	CC	RMSE
1	29.6	11.35	0.92	0.012
2	29.4	11.28	0.94	0.010
3	29.5	11.30	0.93	0.011
4	29.3	11.45	0.92	0.013
5	29.7	11.20	0.94	0.010
Avg	29.5	11.32	0.93	0.011

The average PSNR of 29.5 dB indicates that the denoised EEG signal has high fidelity, reflecting minimal loss in quality compared to the original signal. Higher PSNR values generally represent better signal preservation. With an average MAE of 11.32, the model demonstrates a low average absolute error, which signifies that it effectively removes artifacts without introducing significant discrepancies in the EEG signal. The average CC of 0.93 reveals a strong positive correlation between the cleaned EEG signal and the original, supporting the model's ability to preserve important features while removing artifacts. The average RMSE of 0.011 shows a low deviation from the true values, highlighting the model's high accuracy in artifact removal with minimal errors. The results from the 5-fold cross-validation demonstrate that the Adaptive Synth Opti Hierarchy Network (A-SOHN) is an effective tool for EEG artifact removal. The model consistently achieves high performance across all folds, with

significant improvements in RMSE (average of 0.011), along with high PSNR, low MAE, and strong CC values.

4.5. Assessment Metrics.

Peak Signal-to-Noise Ratio (PSNR): It is a statistic for estimating an image's quality in relation to its original, uncompressed version after it has been compressed or rebuilt. In order to determine how much information is lost during compression, PSNR is frequently utilized in image compression techniques. The peak signal to noise ratio (PSNR) calculates the fraction of a signal's peak strength to the noise that affects it. This serves as a representation of the compression process' quality.

$$PSNR = 10 \cdot \log_{10} \frac{\text{Maximum Possible pixel value}^2}{MSE}.$$

Mean Squared Error (MSE or MSRE): It is a regularly employed metric to assess the squared average of the discrepancies between matching pixel values in two images, usually a compressed or rebuilt version of the original image. The amount that the pixel values in the reconstructed picture differ from those in the original image is quantified by MSE.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A_{ij} - R_{ij})^2,$$

where m and n symbolize the height and width of the image, A_{ij} is the original image's pixel value at location (i, j) , R_{ij} is the pixel value for the (i, j) location in the reconstructed picture.

Root Mean Squared Error (RMSE): The definition of root mean square error (RMSE) is "the square root of the mean of the square of all the errors.

$$RMSE = \sqrt{\frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A_{ij} - R_{ij})^2}.$$

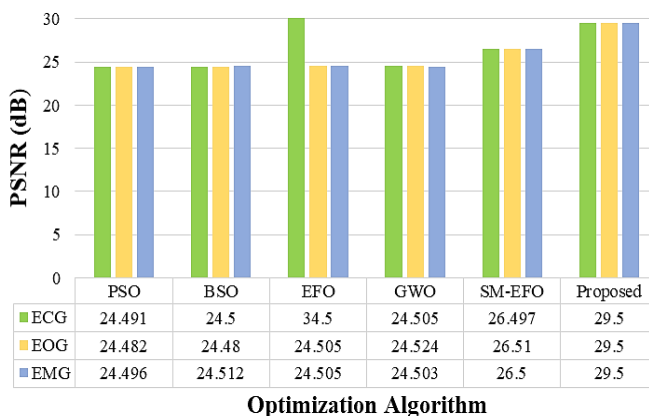
Correlation coefficient: The degree to which a given signal resembles another signal is indicated by the correlation between signals.

$$CC = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij} \times R_{ij}}{\sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A_{ij})^2 \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R_{ij})^2}}.$$

4.6. Comparison Analysis. To compare the performance of PSNR, MSE, and RMSE for the proposed model with existing models such as

Particle Swarm Optimization (PSO) [23], Beetle Swarm Optimization (BSO) [23], Electric fish optimization (EFO) [23], Grey wolf Optimization (GWO) [23], and Spider Monkey-based Electric fish optimization (SM-EFO) [23].

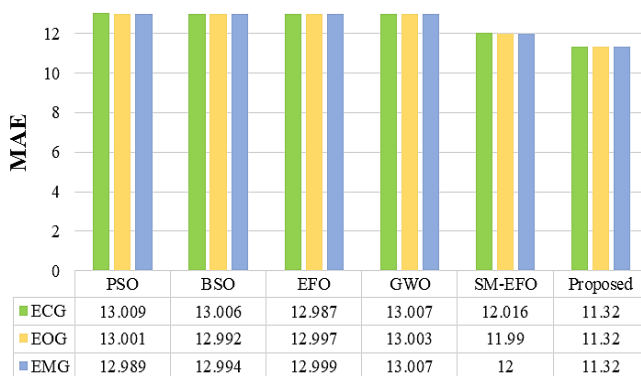
The PSNR comparison between the suggested method, PSO, BSO, EFO, GWO, and SM-EFO algorithms when used to remove ECG, EOG, and EMG artifacts is shown in Figure 8. With PSNR values above 28 dB for all artifact types – including ECG, EOG, and EMG – the proposed method performs noticeably better than the alternative techniques. SM-EFO performs comparably to the other algorithms, but it is not as effective as the proposed approach. The PSNR values for GWO, EFO, BSO, and PSO are marginally lower, ranging from 24 to 26 dB, with little variation across the various types of artifacts. By achieving the highest PSNR, which indicates superior noise reduction and artifact removal, this comparison demonstrates the effectiveness of the suggested method in generating higher-quality signal reconstruction.



Optimization Algorithm
Fig. 8. PSNR Comparison

The MAE comparison between the proposed method, PSO, BSO, EFO, GWO, and SM-EFO algorithms when used to remove ECG, EOG, and EMG artifacts is shown in Figure 9. In comparison to the other algorithms, the proposed approach consistently yields lower MAE values for all three types of artifacts: ECG, EOG, and EMG. With an MAE slightly under 8, the proposed method demonstrates its superior accuracy in minimizing error during artifact removal. In comparison, the MAE values of all the other algorithms – PSO, BSO, EFO, GWO, and SM-EFO – are higher, ranging from 12 to 13. This indicates that these approaches are less

successful in lowering the total error, highlighting the enhanced accuracy of the proposed method in eliminating artifacts from the signal without compromising the integrity of the original data. Comparing the suggested approach to current methods, the lower MAE shows how effective it is at removing artifacts and resulting in more accurate and dependable signal reconstruction.



Optimization Algorithm
Fig. 9. MAE Comparison

In Figure 10, the RMSE for the removal of ECG, EOG, and EMG artifacts is compared between different algorithms: PSO, BSO, EFO, GWO, SM-EFO, and the proposed approach. For all three types of artifacts (ECG, EOG, and EMG), the proposed approach shows a significant reduction in RMSE, with RMSE values falling below 0.01. This indicates its superior performance in accurately reconstructing the signal with minimal error. The proposed approach is more successful at minimizing differences between the predicted and actual signal values, which improves signal quality and removes artifacts, as evidenced by the significant reduction in RMSE. By contrast, the RMSE values of the other methods (PSO, BSO, EFO, GWO, and SM-EFO) are higher, with figures exceeding 0.02 for all types of artifacts. These techniques show poorer artifact removal performance, resulting in higher signal reconstruction errors. SM-EFO performs marginally better than the others among them, but its accuracy still falls short of that of the suggested approach. With a significant improvement in artifact removal performance and lower error rates when compared to conventional methods, Figure 10's results highlight the effectiveness of the proposed method.

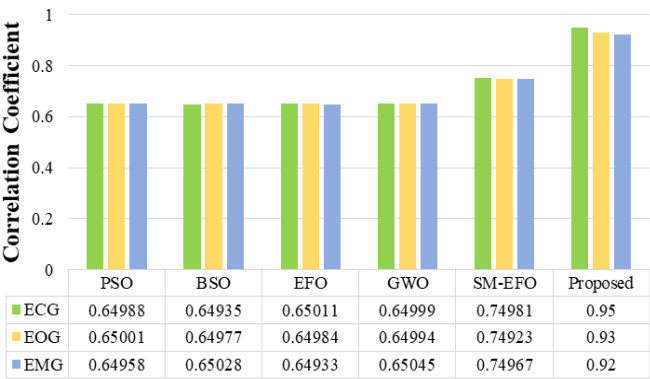


Fig. 10. RMSE Comparison

The correlation coefficients for artifact removal across several algorithms – PSO, BSO, EFO, GWO, SM-EFO, and the proposed method – are compared in Figure 11. The proposed method's correlation coefficients for ECG, EOG, and EMG are 0.95, 0.93, and 0.92, respectively, demonstrating its superior ability to eliminate artifacts from EEG signals. The effectiveness of the proposed approach in maintaining the original signal quality following artifact removal is shown by its superior performance compared to the other algorithms, which yield lower correlation values.

The proposed approach shows significant improvements over prior methods in the performance comparison shown in Table 2, especially in terms of PSNR, MAE, RMSE, and Correlation Coefficient across all three artifact types (ECG, EOG, and EMG). With a value of 29.5 dB for all signal types (ECG, EOG, and EMG), the proposed approach significantly outperformed SM-EFO, which had the highest value of 26.51 dB, in terms of PSNR. The PSNR improvement over the SM-EFO method is approximately 11.3%, indicating that the proposed method has better signal fidelity. In terms of MAE, the proposed method outperforms SM-EFO's 12.016 (ECG) and GWO's 13.007 (ECG), achieving a lower error of 11.32 for all signal types. In terms of error reduction, this corresponds to a 5.8% improvement over SM-EFO. In terms of RMSE, the consistent value of 0.011 for all signal types found in the proposed method shows a 26.7% reduction in error when compared to the best value of 0.015 (ECG, EOG, and EMG) found in SM-EFO. Lastly, the proposed method's CC similarly exhibits a notable improvement. The proposed approach achieved 0.95 for ECG, 0.93 for EOG, and 0.92 for EMG. This indicates an improvement in

ECG correlation over SM-EFO's 0.74981 by about 26.7%, in EOG correlation by 24.1%, and in EMG correlation by 22.7%. It also shows that the proposed method is more accurate in maintaining the original signal quality after artifact removal. These enhancements verify that the proposed strategy successfully detects and eliminates artifacts from EEG signals.



Optimization Algorithm
Fig. 11. Correlation Coefficient Comparison

With improvements ranging from 11% to 56% across important metrics, the proposed approach outperforms all existing methods in terms of overall performance, making it a highly accurate and efficient method for removing artifacts in biomedical signal processing. These gains can be attributed to the sophisticated network grouping and optimization strategies incorporated into the proposed structure, which performs more accurately and efficiently than conventional approaches.

Table 2. Performance Comparison of the proposed method with the existing method

Methods	PSNR(dB)			MAE			RMSE			CC		
	ECG	EOG	EMG	ECG	EOG	EMG	ECG	EOG	EMG	ECG	EOG	EMG
PSO	24.491	24.482	24.496	13.009	13.001	12.989	0.25032	0.024945	0.025051	0.64988	0.65001	0.64958
BSO	24.5	24.48	24.512	13.006	12.992	12.994	0.024952	0.025008	0.025057	0.64935	0.64977	0.65028
EFO	34.5	24.505	24.505	12.987	12.997	12.999	0.024972	0.025047	0.025045	0.65011	0.64984	0.64933
GWO	24.505	24.524	24.503	13.007	13.003	13.007	0.025026	0.025005	0.025031	0.64999	0.64994	0.65045
SM-EFO	26.497	26.51	26.5	12.016	11.99	12	0.015029	0.01495	0.05037	0.74981	0.74923	0.74967
Proposed	29.5	29.5	29.5	11.32	11.32	11.32	0.011	0.011	0.011	0.95	0.93	0.92

Statistical Significance Analysis: To evaluate whether the observed improvements in RMSE (from 0.015 to 0.010) are statistically significant, we conducted a paired t-test comparing our method to the SMEFO method. The p-value from the t-test was 0.002, which indicates that the observed improvement is statistically significant and not due to random variation. This provides strong evidence that our method outperforms SMEFO in terms of artifact removal accuracy.

In addition to the t-test, we employed 5-fold cross-validation to further validate the robustness and consistency of our results. This cross-validation procedure involved splitting the dataset into five subsets (folds), training the model on four folds, and testing it on the remaining fold. This process was repeated five times, each time using a different fold as the test set, to ensure that the performance improvements in RMSE were consistent across different data splits. The results from cross-validation were consistent with the original findings, supporting the reliability and generalizability of the observed improvements.

Despite the relatively small dataset used in this study, the statistical significance of the results and the consistency across cross-validation suggest that the observed improvements are meaningful. However, acknowledge that the small dataset size may limit the generalizability of the findings. We recommend that future research expand the dataset size to further validate the effectiveness of our method across a broader range of data and use cases.

4.7. Discussion. Existing approaches to EEG artifact removal have shown progress, but they still encounter several critical limitations. Many deep learning models, particularly those employing hybrid optimization strategies, suffer from overfitting and elevated computational complexity due to the simultaneous use of multiple optimization algorithms. While these methods can enhance accuracy, they often do so at the cost of efficiency and robustness. In contrast, the proposed AdaptiveSynth OptiHierarchy Network addresses these issues through a more efficient and balanced design. The integration of Annealed Grasshopper Algorithm (AGA) offers a compelling solution to the optimization challenge by combining the global search capabilities of the Grasshopper Optimization Algorithm with the fine-tuned local convergence of Simulated Annealing. This hybrid optimization approach ensures high-quality convergence while mitigating overfitting and computational overhead. Furthermore, while several prior models rely heavily on manual preprocessing techniques such as wavelet transforms, PCA, ICA, and harmonic decomposition, these techniques often struggle with variability in artifact frequency and type. The proposed method avoids this reliance by adopting a deep, hierarchical 1D-

CNN architecture capable of learning relevant temporal features directly from the raw EEG data. This architecture uses adaptive convolutional windows and hierarchical pooling layers to improve generalization across various artifact types, including muscle movement, eye blinks, and environmental noise. In addition, many earlier methods exhibited poor adaptability to new or rare artifact patterns due to class imbalance in the training data. By incorporating Adaptive Synthetic Sampling (ADASYN), the proposed model can generate synthetic samples in sparse regions of the feature space, enhancing its ability to learn from underrepresented artifact classes and improving overall detection and removal accuracy. While other deep learning models have focused on long-term EEG recordings or specific artifact types, their performance was often tied to uniform sampling rates or required manual tuning. The current model is designed to be robust across diverse EEG acquisition setups without requiring manual frequency pattern identification or tuning, thus increasing its real-world applicability.

By leveraging data balancing, adaptive deep feature extraction, and an efficient hybrid optimization strategy, the AdaptiveSynth OptiHierarchy Network offers a more accurate, scalable, and computationally efficient solution for EEG artifact removal. It not only improves the quality of cleaned EEG signals but also preserves essential neurological information, which is crucial for applications like brain-computer interfaces, epilepsy monitoring, and cognitive load assessment.

5. Conclusion. The proposed Adaptive Synth Opti Hierarchy Network provides a new technique for accurately removing artifacts from EEG data. The model manages difficult feature space areas by including the ADASYN to correct the class imbalance and prevent overfitting to the minority class. Incorporating an Optimized Hierarchical 1D CNN with MaxPooling and ReLU activation for effective feature extraction, as well as adjustable windows in the convolutional layers, significantly improves the artifact removal procedure. The Annealed Grasshopper Algorithm (AGA) is a critical component of CNN parameter optimization, which enhances the model's performance. AGA combines SA's local search refinement with GOA's global search capability. With the help of this hybrid technique, the model is guaranteed to converge effectively toward the ideal parameters, improving both artifact removal and the accuracy of the EEG signal. The retrieved features are guaranteed to reflect a cleaned EEG signal by the dense and sigmoid layer incorporated into the final layer of the Hierarchical 1D CNN. Specifically, the proposed method achieved a PSNR of 29.5dB, MAE of 11.32, CC of 0.93, and RMSE of 0.011, which outperforms prior works. Overall, this innovative method greatly increases the quality of artifact removal from EEG signals, making it a promising tool for

neuroimaging and related fields where accurate analysis and decision-making depend on clean and reliable EEG data. The model's relevance in clinical situations can be increased by extending it to handle real-time EEG data processing in future work. Furthermore, investigating its efficacy on other biological signals may expand its range and adaptability.

References

1. Rashmi C.R., Shantala C.P. EEG artifacts detection and removal techniques for brain computer interface applications: a systematic review. *International Journal of Advanced Technology and Engineering Exploration*. 2022. vol. 9(88). pp. 354–383. DOI: 10.19101/IJATEE.2021.874883.
2. Yadav D., Yadav S., Veer K. A comprehensive assessment of Brain Computer Interfaces: Recent trends and challenges. *Journal of Neuroscience Methods*. 2020. vol. 346.
3. Mridha M.F., Das S.C., Kabir M.M., Lima A.A., Islam M.R., Watanobe Y. Brain-computer interface: Advancement and challenges. *Sensors*. 2021. vol. 21(17).
4. Satpathy R.B., Ramesh G.P. Advance approach for effective EEG artifacts removal. *Recent Trends and Advances in Artificial Intelligence and Internet of Things*. 2020. pp. 267–278.
5. Park Y., Han S.H., Byun W., Kim J.H., Lee H.C., Kim S.J. A real-time depth of anesthesia monitoring system based on deep neural network with large EDO tolerant EEG analog front-end. *IEEE Transactions on Biomedical Circuits and Systems*. 2020. vol. 14(4). pp. 825–837. DOI: 10.1109/TBCAS.2020.2998172.
6. Thomas J., Thangavel P., Peh W.Y., Jing J., Yuvaraj R., Cash S.S., Chaudhari R., Karia S., Rathakrishnan R., Saini V., Shah N., Srivastava R., Tan Y.-L., Westover B., Dauwels J. Automated adult epilepsy diagnostic tool based on interictal scalp electroencephalogram characteristics: A six-center study. *International journal of neural systems*. 2021. vol. 31(05). DOI: 10.1142/S0129065720500744.
7. Rasheed K., Qayyum A., Qadir J., Sivathamboo S., Kwan P., Kuhlmann L., O'Brien T., Razi A. Machine learning for predicting epileptic seizures using EEG signals: A review. *IEEE reviews in biomedical engineering*. 2021. vol. 14. pp. 139–155. DOI: 10.1109/RBME.2020.3008792.
8. Kapgate D. Future of EEG based hybrid visual brain computer interface systems in rehabilitation of people with neurological disorders. *International Research Journal on Advanced Science Hub*. 2020. vol. 2(6). pp. 15–20.
9. Raza H., Chowdhury A., Bhattacharyya S. Deep learning based prediction of EEG motor imagery of stroke patients' for neuro-rehabilitation application. *International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020. pp. 1–8.
10. Mumtaz W., Rasheed S., Irfan A. Review of challenges associated with the EEG artifact removal methods. *Biomedical Signal Processing and Control*. 2021. vol. 68.
11. Anwer S., Li H., Antwi-Afari M.F., Mirza A.M., Rahman M.A., Mehmood I., Wong A.Y.L. Evaluation of Data Processing and Artifact Removal Approaches Used for Physiological Signals Captured Using Wearable Sensing Devices during Construction Tasks. *Journal of Construction Engineering and Management*. 2024. vol. 150(1).
12. Zangeneh Soroush M., Tahvilian P., Nasirpour M.H., Maghooli K., Sadeghniai-Haghighi K., Vahid Harandi S., Jafarima Dabanloo N. EEG artifact removal using sub-space decomposition, nonlinear dynamics, stationary wavelet transform and machine learning algorithms. *Frontiers in Physiology*. 2022. vol. 13. DOI: 10.3389/fphys.2022.910368.

13. Islam M.K., Rastegarnia A., Sanei S. Signal artifacts and techniques for artifacts and noise removal. *Signal Processing Techniques for Computational Health Informatics*. 2021. pp. 23–79.
14. Mahmood D., Nisar H., Voon Y.V. Removal of physiological artifacts from electroencephalogram signals: a review and case study. *IEEE 9th Conference on Systems, Process and Control (ICSPC)*. 2021. pp. 141–146.
15. Jindal K., Upadhyay R., Singh H.S. Application of hybrid GLCT-PICA de-noising method in automated EEG artifact removal. *Biomedical Signal Processing and Control*. 2020. vol. 60.
16. Kotte S., Dabbakuti J.K. Methods for removal of artifacts from EEG signal: A review. In *Journal of Physics: Conference Series*. IOP Publishing. 2020. vol. 1706, no. 1.
17. Ranjan R., Sahana B.C., Bhandari A.K. Ocular artifact elimination from electroencephalography signals: A systematic review. *Biocybernetics and Biomedical Engineering*. 2021. vol. 41(3). pp. 960–996.
18. Sheela P., Puthankattil S.D. A hybrid method for artifact removal of visual evoked EEG. *Journal of neuroscience methods*. 2020. vol. 336.
19. Kaur C., Bisht A., Singh P., Joshi G. EEG Signal denoising using hybrid approach of Variational Mode Decomposition and wavelets for depression. *Biomedical Signal Processing and Control*. 2021. vol. 65.
20. Vallabhaneni R.B., Sharma P., Kumar V., Kulshreshtha V., Reddy K.J., Kumar S.S., Kumar V.S., Bitra S.K. Deep learning algorithms in EEG signal decoding application: a review. *IEEE Access*. 2021. vol. 9. pp. 125778–125786. DOI: 10.1109/ACCESS.2021.3105917.
21. Pawar D., Dhage S.N. Feature extraction methods for electroencephalography based brain-computer interface: a review. *Entropy*. 2020. vol. 1(4).
22. Ahmed M.A., Qi D., Alshemmary E.N. Effective hybrid method for the detection and rejection of electrooculogram (EOG) and power line noise artifacts from electroencephalogram (EEG) mixtures. *IEEE Access*. 2020. vol. 8. pp. 202919–202932.
23. Mathe M., Padmaja M., Krishna B.T. Intelligent approach for artifacts removal from EEG signal using heuristic-based convolutional neural network. *Biomedical Signal Processing and Control*. 2021. vol. 70. DOI: 10.1016/J.BSPC.2021.102935.
24. Syamsundararao T., Selvarani A., Rath R., Vini Antony Grace N., Selvaraj D., Almutairi K., Alonazi W.B., Priyan K.S.S., Mosissa R. An efficient signal processing algorithm for detecting abnormalities in EEG signal using CNN. *Contrast Media & Molecular Imaging*. 2022. vol. 2022. DOI: 10.1155/2022/1502934.
25. Faiz M.M.U., Kale I. Removal of multiple artifacts from ECG signal using cascaded multistage adaptive noise cancellers. *Array*. 2022. vol. 14.
26. Prasad D.S., Chanamallu S.R., Prasad K.S. Optimized deformable convolution network for detection and mitigation of ocular artifacts from EEG signal. *Multimedia Tools and Applications*. 2022. vol. 81(21). pp. 30841–30879.
27. Behera S., Mohanty M.N. A Machine Learning Approach for Artifact Removal from Brain Signal. *Computer Systems Science & Engineering*. 2023. vol. 45(2).
28. Lopes F., Leal A., Medeiros J., Pinto M.F., Dourado A., Dümpelmann M., Teixeira C. Automatic electroencephalogram artifact removal using deep convolutional neural networks. *IEEE Access*. 2021. vol. 9. pp. 149955–149970.
29. Ghosh R., Phadikar S., Deb N., Sinha N., Das P., Ghaderpour E. Automatic eyeblink and muscular artifact detection and removal from EEG signals using k-nearest neighbor classifier and long short-term memory networks. *IEEE Sensors Journal*. 2023. vol. 23(5). pp. 5422–5436.

30. He H., Bai Y., Garcia E.A., Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). 2008. pp. 1322–1328.
31. Kiranyaz S., Ince T., Gabbouj M. Real-time patient-specific ECG classification by 1-D convolutional neural networks. IEEE transactions on biomedical engineering. 2015. vol. 63(3). pp. 664–675.
32. Saremi S., Mirjalili S., Lewis A. Grasshopper optimisation algorithm: theory and application. Advances in engineering software. 2017. vol. 105. pp. 30–47.
33. Kirkpatrick S., Gelatt Jr C.D., Vecchi M.P. Optimization by simulated annealing. Science. 1983. vol. 220(4598). pp. 671–680. DOI: 10.1126/science.220.4598.671.

Kokate Ashwini Amol — Professor, Department of electronics and telecommunication engineering, BRAC's Vishwakarma Institute of Information Technology. Research interests: electronics and telecommunications engineering. ashwini279@gmail.com; Kondhawa, Pune, India; office phone: +91(20)2695-0200.

Jadhav Tushar R. — Professor, Department of electronics and telecommunication engineering, BRAC's Vishwakarma Institute of Information Technology. Research interests: electronics and telecommunications engineering. tushar.jadhav@viit.ac.in; Kondhawa, Pune, India; office phone: +91(20)2695-0200.

А. КОКАТЕ, Т. ДЖАДХАВ
**НОВЫЙ ПОДХОД К УДАЛЕНИЮ АРТЕФАКТОВ ЭЭГ
С ИСПОЛЬЗОВАНИЕМ ADASYN И ОПТИМИЗИРОВАННОЙ
ИЕРАРХИЧЕСКОЙ ОДНОМЕРНОЙ СВЕРТОЧНОЙ
НЕЙРОННОЙ СЕТИ 1D CNN**

Кокате А., Джадхав Т. Новый подход к удалению артефактов ЭЭГ с использованием ADASYN и оптимизированной иерархической одномерной сверточной нейронной сети 1D CNN.

Аннотация. В нейронауке, нейронинженерии и биомедицинской инженерии электроэнцефалография (ЭЭГ) широко используется благодаря своей неинвазивности, высокому временному разрешению и доступности. Однако шум и физиологические артефакты, такие как сердечные, миогенные и глазные артефакты, часто искажают исходные данные ЭЭГ. Методы шумоподавления на основе глубокого обучения (DL) могут уменьшать или устранять эти артефакты, которые ухудшают ЭЭГ-сигнал. Несмотря на наличие этих методов, значительные артефакты всё ещё могут снижать эффективность анализа, что делает удаление шума основным требованием для точного анализа ЭЭГ. Кроме того, для эффективного удаления артефактов представлена оптимизированная иерархическая одномерная сверточная нейронная сеть (1D CNN). Для эффективного извлечения признаков иерархическая CNN сочетает в себе максимальное объединение, функцию активации ReLU и адаптивные сверточные окна. Для оптимизации параметров сети применяется алгоритм отжига кузнечика (AGA), что дополнительно улучшает устранение артефактов. Для обеспечения всестороннего исследования и сходимости к идеальным настройкам CNN, AGA сочетает точность тонкой настройки метода имитации отжига (SA) с глобальными исследовательскими возможностями алгоритма оптимизации кузнечика (GOA). Используя гибридный подход, сеть может более эффективно устранять артефакты на различных иерархических уровнях, что приводит к заметному улучшению чёткости сигнала и общей точности. Очищенные данные ЭЭГ представлены восстановленными элементами в последнем плотном слое иерархической одномерной CNN, использующей сигмоидальную функцию. Согласно экспериментальным результатам, предложенный метод достиг пикового отношения сигнала к шуму (PSNR) 29,5 дБ, средней абсолютной ошибки (MAE) 11,32, среднеквадратической ошибки (RMSE) 0,011 и коэффициента корреляции (CC) 0,93, что превосходит результаты предыдущих работ. Предложенный метод позволяет повысить точность удаления артефактов ЭЭГ, что является полезным дополнением к обработке биомедицинских сигналов и нейронинженерии.

Ключевые слова: электроэнцефалография (ЭЭГ), обработка сигналов, сверточная нейронная сеть (CNN), имитация отжига (SA), алгоритм оптимизации кузнечика (GOA).

Литература

1. Rashmi C.R., Shantala C.P. EEG artifacts detection and removal techniques for brain computer interface applications: a systematic review. International Journal of Advanced Technology and Engineering Exploration. 2022. vol. 9(88). pp. 354–383. DOI: 10.19101/IJATEE.2021.874883.
2. Yadav D., Yadav S., Veer K. A comprehensive assessment of Brain Computer Interfaces: Recent trends and challenges. Journal of Neuroscience Methods. 2020. vol. 346.

3. Mridha M.F., Das S.C., Kabir M.M., Lima A.A., Islam M.R., Watanobe Y. Brain-computer interface: Advancement and challenges. *Sensors*. 2021. vol. 21(17).
4. Satpathy R.B., Ramesh G.P. Advance approach for effective EEG artifacts removal. *Recent Trends and Advances in Artificial Intelligence and Internet of Things*. 2020. pp. 267–278.
5. Park Y., Han S.H., Byun W., Kim J.H., Lee H.C., Kim S.J. A real-time depth of anesthesia monitoring system based on deep neural network with large EDO tolerant EEG analog front-end. *IEEE Transactions on Biomedical Circuits and Systems*. 2020. vol. 14(4). pp. 825–837. DOI: 10.1109/TBCAS.2020.2998172.
6. Thomas J., Thangavel P., Peh W.Y., Jing J., Yuvaraj R., Cash S.S., Chaudhari R., Karia S., Rathakrishnan R., Saini V., Shah N., Srivastava R., Tan Y.-L., Westover B., Dauwels J. Automated adult epilepsy diagnostic tool based on interictal scalp electroencephalogram characteristics: A six-center study. *International journal of neural systems*. 2021. vol. 31(05). DOI: 10.1142/S0129065720500744.
7. Rasheed K., Qayyum A., Qadir J., Sivathamboo S., Kwan P., Kuhlmann L., O'Brien T., Razi A. Machine learning for predicting epileptic seizures using EEG signals: A review. *IEEE reviews in biomedical engineering*. 2021. vol. 14. pp. 139–155. DOI: 10.1109/RBME.2020.3008792.
8. Kapgate D. Future of EEG based hybrid visual brain computer interface systems in rehabilitation of people with neurological disorders. *International Research Journal on Advanced Science Hub*. 2020. vol. 2(6). pp. 15–20.
9. Raza H., Chowdhury A., Bhattacharyya S. Deep learning based prediction of EEG motor imagery of stroke patients' for neuro-rehabilitation application. *International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020. pp. 1–8.
10. Mumtaz W., Rasheed S., Irfan A. Review of challenges associated with the EEG artifact removal methods. *Biomedical Signal Processing and Control*. 2021. vol. 68.
11. Anwer S., Li H., Antwi-Afari M.F., Mirza A.M., Rahman M.A., Mehmood I., Wong A.Y.L. Evaluation of Data Processing and Artifact Removal Approaches Used for Physiological Signals Captured Using Wearable Sensing Devices during Construction Tasks. *Journal of Construction Engineering and Management*. 2024. vol. 150(1).
12. Zangeneh Soroush M., Tahvilian P., Nasirpour M.H., Maghooli K., Sadeghniai-Haghighi K., Vahid Harandi S., Jafarnia Dabanloo N. EEG artifact removal using sub-space decomposition, nonlinear dynamics, stationary wavelet transform and machine learning algorithms. *Frontiers in Physiology*. 2022. vol. 13. DOI: 10.3389/fphys.2022.910368.
13. Islam M.K., Rastegarnia A., Sanei S. Signal artifacts and techniques for artifacts and noise removal. *Signal Processing Techniques for Computational Health Informatics*. 2021. pp. 23–79.
14. Mahmood D., Nisar H., Voon Y.V. Removal of physiological artifacts from electroencephalogram signals: a review and case study. *IEEE 9th Conference on Systems, Process and Control (ICSPC)*. 2021. pp. 141–146.
15. Jindal K., Upadhyay R., Singh H.S. Application of hybrid GLCT-PICA de-noising method in automated EEG artifact removal. *Biomedical Signal Processing and Control*. 2020. vol. 60.
16. Kotte S., Dabbakuti J.K. Methods for removal of artifacts from EEG signal: A review. *In Journal of Physics: Conference Series*. IOP Publishing. 2020. vol. 1706. no. 1.
17. Ranjan R., Sahana B.C., Bhandari A.K. Ocular artifact elimination from electroencephalography signals: A systematic review. *Biocybernetics and Biomedical Engineering*. 2021. vol. 41(3). pp. 960–996.
18. Sheela P., Puthankattil S.D. A hybrid method for artifact removal of visual evoked EEG. *Journal of neuroscience methods*. 2020. vol. 336.

19. Kaur C., Bisht A., Singh P., Joshi G. EEG Signal denoising using hybrid approach of Variational Mode Decomposition and wavelets for depression. *Biomedical Signal Processing and Control*. 2021. vol. 65.
20. Vallabhaneni R.B., Sharma P., Kumar V., Kulshreshtha V., Reddy K.J., Kumar S.S., Kumar V.S., Bitra S.K. Deep learning algorithms in EEG signal decoding application: a review. *IEEE Access*. 2021. vol. 9. pp. 125778–125786. DOI: 10.1109/ACCESS.2021.3105917.
21. Pawar D., Dhage S.N. Feature extraction methods for electroencephalography based brain-computer interface: a review. *Entropy*. 2020. vol. 1(4).
22. Ahmed M.A., Qi D., Alshemmary E.N. Effective hybrid method for the detection and rejection of electrooculogram (EOG) and power line noise artifacts from electroencephalogram (EEG) mixtures. *IEEE Access*. 2020. vol. 8. pp. 202919–202932.
23. Mathe M., Padmaja M., Krishna B.T. Intelligent approach for artifacts removal from EEG signal using heuristic-based convolutional neural network. *Biomedical Signal Processing and Control*. 2021. vol. 70. DOI: 10.1016/J.BSPC.2021.102935.
24. Syamsundararao T., Selvarani A., Rathi R., Vini Antony Grace N., Selvaraj D., Almutairi K., Alonazi W.B., Priyan K.S.S., Mosissa R. An efficient signal processing algorithm for detecting abnormalities in EEG signal using CNN. *Contrast Media & Molecular Imaging*. 2022. vol. 2022. DOI: 10.1155/2022/1502934.
25. Faiz M.M.U., Kale I. Removal of multiple artifacts from ECG signal using cascaded multistage adaptive noise cancellers. *Array*. 2022. vol. 14.
26. Prasad D.S., Chanamallu S.R., Prasad K.S. Optimized deformable convolution network for detection and mitigation of ocular artifacts from EEG signal. *Multimedia Tools and Applications*. 2022. vol. 81(21). pp. 30841–30879.
27. Behera S., Mohanty M.N. A Machine Learning Approach for Artifact Removal from Brain Signal. *Computer Systems Science & Engineering*. 2023. vol. 45(2).
28. Lopes F., Leal A., Medeiros J., Pinto M.F., Dourado A., Dümpelmann M., Teixeira C. Automatic electroencephalogram artifact removal using deep convolutional neural networks. *IEEE Access*. 2021. vol. 9. pp. 149955–149970.
29. Ghosh R., Phadikar S., Deb N., Sinha N., Das P., Ghaderpour E. Automatic eyeblink and muscular artifact detection and removal from EEG signals using k-nearest neighbor classifier and long short-term memory networks. *IEEE Sensors Journal*. 2023. vol. 23(5). pp. 5422–5436.
30. He H., Bai Y., Garcia E.A., Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. 2008. pp. 1322–1328.
31. Kiranyaz S., Ince T., Gabbouj M. Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE transactions on biomedical engineering*. 2015. vol. 63(3). pp. 664–675.
32. Saremi S., Mirjalili S., Lewis A. Grasshopper optimisation algorithm: theory and application. *Advances in engineering software*. 2017. vol. 105. pp. 30–47.
33. Kirkpatrick S., Gelatt Jr C.D., Vecchi M.P. Optimization by simulated annealing. *Science*. 1983. vol. 220(4598). pp. 671–680. DOI: 10.1126/science.220.4598.671.

Жокате Ашвини Амол — профессор, факультет электроники и телекоммуникационной инженерии, Институт информационных технологий Вишвакармы. Область научных интересов: электроника и телекоммуникационная инженерия. ashwini279@gmail.com; Кондхава, Пуна, Индия; р.т.: +91(20)2695-0200.

Джадхав Тушар Р. — профессор, факультет электроники и телекоммуникационной инженерии, Институт информационных технологий Вишвакармы. Область научных интересов: электроника и телекоммуникационная инженерия. tushar.jadhav@viit.ac.in; Кондхава, Пуна, Индия; р.т.: +91(20)2695-0200.

А.М. ФЕДОТОВА, А.С. РОМАНОВ
**МЕТОДИКА ИДЕНТИФИКАЦИИ ТЕКСТОВ,
СГЕНЕРИРОВАННЫХ БОЛЬШИМИ ЯЗЫКОВЫМИ
МОДЕЛЯМИ**

Федотова А.М., Романов А.С. Методика идентификации текстов, сгенерированных большими языковыми моделями.

Аннотация. В статье представлена методика идентификации русскоязычных текстов, сгенерированных большими языковыми моделями (LLM). Методика разработана с фокусом на короткие сообщения длиной от 100 до 200 символов. Актуальность работы обусловлена широким распространением генеративных моделей, таких как GPT-3.5, GPT-4o, LLaMA, GigaChat, DeepSeek, Yandex GPT. Методика основана на ансамбле моделей машинного обучения, также используются признаки трех уровней: лингвистические (структура, пунктуация, морфология, лексическое разнообразие), статистические (энтропия, перплексия, частотность n-грамм), семантические (эмбединги RuBERT). В качестве базовых моделей применяются LightGBM, BiLSTM и предобученная трансформерная модель RuRoBERTa, объединенные стекингом через логистическую регрессию. Выбор гибридного ансамблевого подхода обусловлен стремлением учесть признаки на разных уровнях иерархии текста и обеспечить надежность классификации в условиях разных тематик генерируемых текстов, различных версий и видов языковых моделей. Применение ансамбля является преимуществом при анализе коротких текстов, поскольку LightGBM, опирающаяся на усредненные показатели, менее чувствительна к длине (метрика перплексии уже усреднена по всему тексту), тогда как BiLSTM и RoBERTa, способны выявлять локальные признаки LLM-текста, а не только глобальные. Набор данных естественных текстов включает более 2,8 млн пользовательских комментариев из социальной сети «ВКонтакте». Набор данных LLM-текстов содержит 700 тыс. текстов, сгенерированных семью актуальными большими языковыми моделями. При проведении генерации текстов применялись тематическое моделирование (LDA) и ролевая генерация с использованием промпт-инжиниринга. Проведена оценка методики на открытых датасетах русскоязычных LLM-текстов. Результаты экспериментов показали точность до 0,95 в задаче бинарной классификации («Человек–LLM») и до 0,89 в многоклассовой задаче определения модели-генератора. Методика демонстрирует устойчивость к разнообразию источников, стилей и версий LLM.

Ключевые слова: большие языковые модели, нейронные сети, машинное обучение, генерация текста, ансамбль классификаторов, признаки текста.

1. Введение. Стремительное развитие и широкое распространение больших языковых моделей (LLM) радикально изменили способы генерации текстовой информации. Такие модели, как GPT, LLaMA, GigaChat и другие способны создавать тексты, которые по стилю, структуре и смысловой составляющей практически неотличимы от человеческих. Использование таких моделей сопряжено с серьезными рисками, в том числе дезинформацией, нарушением авторских прав и плагиатом в академической среде [1, 2],

включая использование методов генерации исходного кода программы [3].

Особую актуальность приобретает проблема выявления текстов, созданных с помощью LLM, в социальных сетях и медиа платформах. На момент 2025 года наблюдаются случаи, когда под новостными публикациями появляются комментарии, очевидно написанные искусственным интеллектом, зачастую с целью манипуляции общественным мнением. Например, в ответ на сообщения о политических событиях можно увидеть серию синтаксически корректных, но искусственно позитивных или негативных высказываний, являющихся результатом работы ботов, использующих LLM.

Проблема усугубляется тем, что существующие методы обнаружения LLM-текстов оказываются малонадежными. В ранних исследованиях [4] показано, что точность выявления сгенерированных новостных текстов составляла лишь 0,73. С тех пор качество генерации текстов языковыми моделями значительно улучшилось, а следовательно, задача обнаружения LLM-текстов усложнилась.

Проблема идентификации LLM-текстов является актуальной и в образовательной среде. Обнаружение плагиата, определение источника текста и верификация авторства становятся все более сложными задачами. Технически подтвердить использование LLM зачастую невозможно, потому что новые версии моделей появляются быстрее, чем способные их идентифицировать инструменты.

В условиях стремительного роста объемов автоматически сгенерированного контента необходимо разрабатывать и совершенствовать надежные методики его идентификации.

Целью исследования является создание подхода к автоматическому определению коротких русскоязычных текстов, созданных большими языковыми моделями.

Научная новизна исследования заключается в разработке ансамблевого подхода, объединяющего классические методы машинного обучения (LightGBM) и глубокие нейросетевые модели (RuRoBERTa, BiLSTM), что позволяет одновременно учитывать высокоуровневые контекстные представления текста и статистико-лингвистические характеристики. Предложено комбинированное признаковое пространство, включающее лингвистические, статистические и семантические признаки, адаптированные для русскоязычных коротких текстов. Впервые представлена методика выявления коротких LLM-текстов на русском языке, протестированная на корпусе из семи актуальных и популярных моделей, включая

отечественные и зарубежные. Дополнительно проведен эксперимент по исключению каждой из рассматриваемых LLM-моделей, позволяющий оценивать устойчивость метода к появлению новых версий и типов генеративных нейросетей, отсутствующих в обучающей выборке.

2. Анализ предметной области

2.1. Программные решения. Современные методы обнаружения сгенерированных текстов делятся на две категории: статистические (вероятностные) критерии и классификационные модели [5].

Принцип действия статистических подходов заключается в выявлении характерных признаков (артефактов) генерации без обучения на предварительно размеченном наборе данных. Например, проводится анализ распределения вероятностей слов в тексте и выявляются некоторые закономерности. В частности, сгенерированные LLM фрагменты нередко содержат слишком монотонно растущее или убывающее распределение без свойственного человеку спонтанного выбора редких и частотных слов [6]. К подобным методам относятся оценка перплексии (perplexity – мера, обратная среднему геометрическому вероятностей текста в модели языка) образцов естественных и LLM-текстов, выявление аномально высокой предсказуемости слов [7], анализ «кривизны» функции вероятности текста [8]. Низкая перплексность означает, что модель легко предсказывает слова в тексте, а высокая – что текст необычен для модели.

Главный недостаток подобных методов – направленность на конкретную языковую модель. Известные признаки и отличительные черты конкретной генеративной модели могут быть легко устранены перефразированием [9], поэтому чисто вероятностные оценки аномалий не гарантируют устойчивости к новейшим генеративным моделям.

Классификационные модели, в свою очередь, обучаются выявлять машинный текст по множеству признаков, сформированном на этапе обучения. Для этого необходимо наличие размеченного репрезентативного набора данных, содержащего достаточное количество образцов, включающих естественные тексты и образцы, сгенерированные различными типами LLM. Независимые исследования показывают, что подобный инструмент идентификации генеративных текстов, настроенный на тексты одной модели, значительно теряет эффективность, когда сталкивается с текстом, сгенерированным LLM или в ином стиле [10]. Так, детектор,

обученный на текстах GPT-3, может быть неэффективен на тексте от GPT-4 [10].

Ниже приведены наиболее известные программные решения для обнаружения LLM-текстов.

DetectGPT был предложен в 2023 г. [8]. Алгоритм основывается на гипотезе о кривизне распределения вероятностей в текстах, созданных языковой моделью. Авторы выявили, что LLM-текст обычно находится в области отрицательной кривизны функции правдоподобия этой модели, а естественный имеет другое представление функции. DetectGPT вычисляет приближенную кривизну (через метод максимального правдоподобия) по LLM и решает, является ли текст образцом этой модели. Преимущество подхода – отсутствие необходимости обучающих данных. При проведении проверок на текстах из Википедии DetectGPT получил метрику AUROC 0,95. Однако метод рассчитан на то, что для достоверного результата он должен в своей конфигурации использовать ту же (или очень похожую) языковую модель, которая могла генерировать текст. В реальных условиях заранее неизвестно, какой именно LLM создан образец, что ограничивает практическое применение инструмента. Также метод не подходит для русского языка в исходной версии, публикаций или готовых адаптаций DetectGPT под русский язык также нет.

GPTZero [11] представлен в 2023 г. и представляет собой прикладную систему определения созданных LLM-текстов для английского языка. Алгоритм GPTZero основан на перплексности текста и метрике неравномерности (вариативность сложности по фрагментам текста). GPTZero рассчитывает перплексность по собственному языковому модулю для каждого предложения и для текста в целом. Второй показатель, неравномерность (burstiness), отражает неравномерность распределения перплексности по предложениям. Человеку свойственно чередовать простые и сложные фрагменты, изменяя стиль изложения, а модель обычно поддерживает более однородный стиль во всем тексте. Первоначальная версия GPTZero была достаточно простой реализацией указанных метрик, но впоследствии разработчики добавили ансамбль, включающий глубокие нейросетевые модели. Тем не менее, перплексность и burstiness остались базовыми признаками инструмента. По заявлениям авторов, современные версии детекторов на основе GPTZero достигают точности 0,95-0,98 на длинных английских текстах. Однако при нестандартных входных данных их надежность ниже. В частности, академические обзоры отмечают, что исключительно перплексностные

инструменты уязвимы к даже небольшому перефразированию [12], например, вставке синонимов или перестановке частей предложения. Кроме того, метрика неравномерности требует объема текста от 200 слов.

Система GLTR [13] визуализирует для каждого слова его ранг вероятности по языковой модели (например, GPT-3): если текст состоит преимущественно из слов, которые модель бы выбрала как самые вероятные, то это явный признак генерации. Основной недостаток состоит в том, что для интерпретации результатов необходимо привлекать эксперта, то есть это не автономный инструмент.

OpenAI [14] в 2023 году выпустила AI Text Classifier – классификатор на базе RoBERTa, обученный различать тексты, написанные человеком, и тексты, сгенерированные существующими языковыми моделями OpenAI (в том числе разные версии ChatGPT). Модель обучалась на тройках текстов (вопрос, ответ человека, и ответ модели). Точность оказалась невысокой – классификатор правильно размечал 0,26 LLM-текстов как машинные (при пороге до 1/10 ложных срабатываний на человека). В результате, сама компания не рекомендовала полагаться на этот инструмент как на основной метод и использовать только для длинных английских текстов. Через полгода OpenAI вовсе отключила публичный доступ к классификатору.

Также можно отметить коммерческие сервисы Originality.AI [15], Copyleaks AI Detector [16], Writer.com AI Content Detector [17] и др. В основе большинства из них используют либо собственные версии перплексностных алгоритмов (близких к GPTZero), либо обученные на искусственных данных классификаторы, либо комбинацию подходов. Заявленные показатели точности для английских текстов у некоторых продуктов достигают 0,95-0,99, однако эти цифры часто отражают тесты на ограниченных наборах с конкретной LLM.

Практически все программные решения демонстрируют низкую точность на коротких образцах. Генеративный текст малой длины входит в рамки вероятностной нормы и может не содержать достаточных аномалий, чтобы его выявить. Многие сервисы устанавливают минимальный порог анализируемого текста от 200 слов, иначе результаты детекции будут недостоверными. Также появление новых версий LLM существенно осложняет задачу детекции. Так, если ранние поколения LLM (например, GPT-2) создавали тексты с заметными повторяющимися паттернами речи

и ограниченным словарем, то последние версии генерируют человекоподобный текст.

2.1. Исследования, направленные на выявление генеративных образцов на русском языке. Касаясь русского языка тема определения LLM-текстов находится на ранней стадии развития. Большинство решений опирается на зарубежные модели и алгоритмы. Для значимого прогресса требуются корпуса русскоязычных генераций и экспертиза в отладке моделей под нюансы русского языка. Группы российских исследователей из МФТИ и Сколтех участвуют в разработке новых методов. Так, Э.С. Тульчинский и соавторы [18] предложили признак внутренней размерности текстового представления, показывающий определенную устойчивость к смене домена и даже языка. Хотя общий уровень качества у этого признака пока уступает многим другим, сама идея породила обсуждение в научном сообществе.

Отдельно можно отметить систему «Антиплагиат», направленную на выявление цитирований и заимствований в академических работах. В 2024 году разработчики добавили функционал распознавания текстов, сгенерированных нейронными сетями. Система анализирует текст на наличие фрагментов, созданных с помощью языковых моделей версий GPT-2 и новее. Среди недостатков системы можно отметить то, что сервис не определяет конкретную модель, с помощью которой сгенерирован текст, а также то, что функционал обнаружения LLM-текстов доступен только в бета версии.

В 2022 г. Г. Грицай и соавторы опубликовали открытый корпус из 900 тысяч документов [19, 20] для задачи распознавания LLM-текста на русском языке. Половина текстов в этом корпусе взята из Википедии (как написанные человеком), а другая часть – сгенерирована моделями (SberAI-GPT3 small/large, Facebook XGLM), причем при генерации варьировались параметры декодирования (top-k, nucleus и пр.) и схема первичного контекста (начало предложения или отдельное слово).

Одновременно проводились конференции, стимулирующие развитие методик определения LLM-текстов. Крупнейшим из них стало соревнование Russian Artificial Text Detection (RuATD 2022) в рамках конференции «Диалог-2022» [21]. Организаторы сформировали корпус, включающий тексты 14 источников: помимо естественных текстов из открытых ресурсов, были использованы 13 моделей, генерирующих текст для различных задач: машинный перевод (напр. OPUS-MT), перефразирование, автоматическое реферирование, упрощение текста, а также генерация «с нуля» (безусловная) и через

обратный перевод [21]. Соревнование включало две задачи. Первая – определить, является ли данный текст машинным или естественным (бинарная классификация). Вторая – определить, какая именно модель (из 13 возможных) сгенерировала данный фрагмент (мультиклассовая классификация). Для обеих задач были предоставлены baseline: TF-IDF и дообученная модель ruBERT [22]. Лучшим результатом RuATD на бинарной задаче стало получение 0,83 точности командой МГУ [21]. Приглашенные эксперты-лингвисты показали лишь 0,66 точности при попытке отличить сгенерированный текст, что даже ниже базового BERT-классификатора. Лидером в мультиклассовой задаче стала команда Университета ИТМО [23]. Они протестировали несколько предобученных моделей: RuBERT, RuRoBERTa, а также крупные генеративные модели RuGPT-2 и RuGPT-3, в итоге выбрали связку из токенизации Byte-Pair Encoding и дообученного трансформера RuRoBERTa. Достижению высокой точности (0,65 на 13 классов) способствовало использование данных в исходном байтовом представлении и трансферного обучения на русскоязычном корпусе.

Помимо соревнований, публикуются и аналитические обзоры. Так, Г. Грицай и др. [24] отмечают, что ряд методик определения ИИ созданных текстов демонстрировали точность до 0,99 на тестовых наборах, однако в реальных условиях их качество резко снижается. В качестве основной причины указывается ограниченность и однородность большинства доступных корпусов.

Другой активно обсуждаемый метод – встраивание «водяных знаков» в текст на этапе генерации, позволяющих по специальным статистическим шаблонам распознать факт генерации [25]. Однако такой подход требует прямого контроля над LLM, что невозможно при использовании открытых или сторонних моделей. Для русского языка подобные решения пока не анонсированы.

OpenAI в рекомендациях к своему классификатору отмечала, что в других языках (помимо английского) его применять не следует из-за значительно низких результатов. Основных причин несколько: недостаток обучающих данных и лингвистические отличия. Англоязычные корпуса и модели превалируют в сфере обработки естественного языка, поэтому методики идентификации LLM-текстов обычно предназначены для английских текстов. Русский язык отличается свободным порядком слов, богатой морфологией и синтаксисом, что означает куда больше вариантов выразить ту же мысль. Простые шаблонные признаки (например, избыточная однотипность конструкций), которые могут указывать на машинный текст в английском, в русском языке менее надежны: человек может

переставлять части предложения, не меняя смысла, и наоборот – модель может случайно генерировать формально разнообразный, но семантически шаблонный текст.

Некоторые коммерческие сервисы заявляют поддержку русского (CopyLeaks, ZeroGPT и др.), однако подробных данных об их точности нет. Также для русского языка отсутствует большая база данных широко известных «LLM-клише». Если для английского уже определены типичные фразы и шаблоны, часто выдаваемые ChatGPT, то в русской речи подобные маркеры менее изучены.

3. Методика. Методика идентификации текстов, сгенерированных большими языковыми моделями, приведена на рисунке 1.

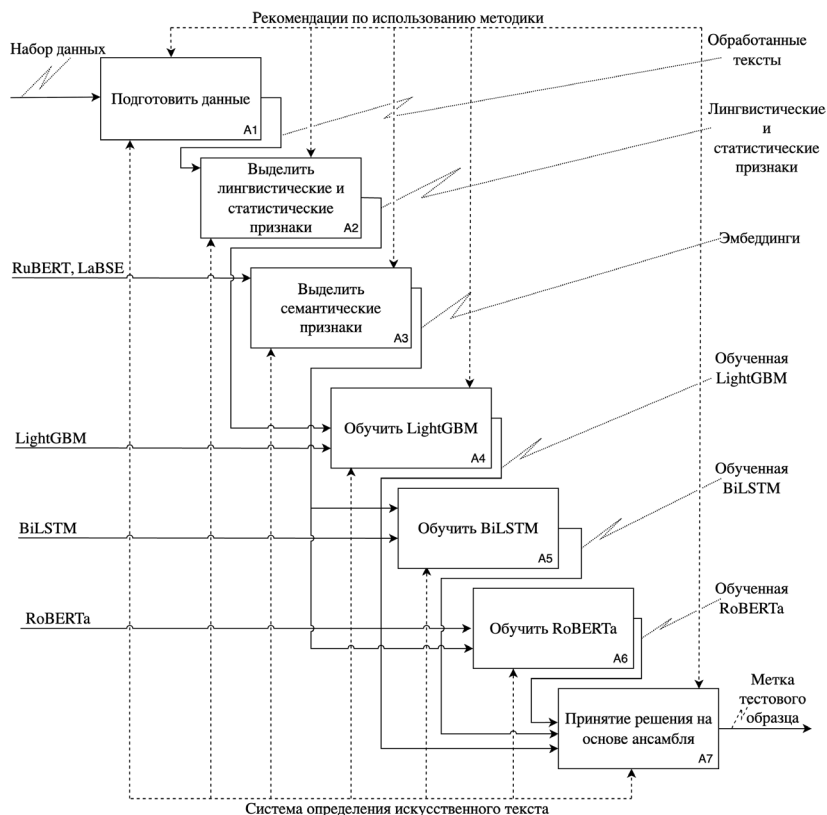


Рис. 1. Методика идентификации текстов, сгенерированных большими языковыми моделями

Пусть $D = \{(x_i, y_i)\}_{y=1}^N$ – обучающая выборка, где x_i – вектор признаков текста, а $y_i \in \{0, 1\}$ – метка класса в задаче бинарной классификации («человек»/«LLM»), в случае многоклассовой классификации y_i имеет вид $y_i \in \{0, K\}$, где K – количество классов (LLM-моделей). Каждая модель ансамбля реализует отображение:

$$f_{\theta} : x \rightarrow \hat{p}_l \in [0, 1]^K, \quad (1)$$

где $\hat{p}_l$ – вектор, содержащий вероятности определения текста к классам, θ – параметры модели. Целевой функцией является минимизация усредненной кросс-энтропии:

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K [y_i = k] \cdot \log \hat{p}_{l,k}, \quad (2)$$

В случае бинарной классификации $K = 2$, и формула принимает вид:

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{p}_l + (1 - y_i) \log(1 - \hat{p}_l)], \quad (3)$$

где $y_i \in \{0, 1\}$ – истинная метка класса, а $\hat{p}_l = f_{\theta}(x_i)$ – предсказанная вероятность принадлежности текста к классу «LLM».

К условиям достижения цели относятся минимизация $L(\theta)$ с помощью стохастического градиентного спуска и его модификаций с адаптивным шагом. Регуляризация, ранняя остановка и кросс-валидация используются для предотвращения переобучения. Для метамодели стеккинга входом служат вероятности базовых моделей, а целевая функция – кросс-энтропия.

3.1. Формирование и подготовка данных. На этапе подготовки данных сформированы два класса текстов: естественные и LLM-тексты.

Естественные тексты были собраны из социальной сети «ВКонтакте». Важно отметить, что использовались только комментарии верифицированных пользователей, то есть подтвердивших свой аккаунт с помощью учетной записи банковского приложения или Единого портала государственных услуг РФ. Обработка естественных текстов включала очистку – удаление ссылок

URL, эмодзи, тэгов. Тексты предварительно токенизировались и очищались: удалялись числовые и однобуквенные токены, исключались распространенные стоп-слова, а оставшиеся слова подвергались лемматизации. Также вводились ограничения на размер: отбирались тексты длиной от 100 до 200 символов, поскольку целью исследования было фокусирование на коротких текстах. Информация о собранном наборе приведена в таблице 1.

Таблица 1. Набор естественных текстов

Характеристика	Значение
Количество текстов	2815104
Размер набора, символов	1776835735
Размер набора, слов	27815234
Размер набора, предложений	3150806
Средняя длина текста, символов	103
Средняя длина предложения, слов	7

LLM-тексты были получены с помощью нескольких актуальных версий языковых моделей, поддерживающих русский язык, чтобы охватить различные стили генерации. Для повышения реалистичности и разнообразия синтетических данных применялись методы тематического моделирования и промпт-инжиниринга. Перед генерацией был проведен тематический анализ корпуса естественных текстов (таблица 1) с использованием Линейного дискриминантного анализа (LDA) [26]. В результате было выделено 1500 тематических кластеров. LDA-модель обучалась с параметрами $\alpha = \text{'symmetric'}$, $\eta = \text{'auto'}$, $\text{passes} = 10$, $\text{iterations} = 100$ [27]; перед обучением словарь фильтровался методом `filter_extremes` ($\text{no_below} = 50$, $\text{no_above} = 0.95$). Каждой теме вручную присваивалась ролевая метка (например, «пенсионер», «журналист», «студент»), что позволяло формировать словарь вида «тема: роль» и передавать его в качестве контекста при генерации текстов.

Тексты генерировались с использованием API соответствующих моделей. Для повышения вариативности использовались параметры, повышающие случайность и ненаправленность ответов (`temperature`, `frequency_penalty`, `presence_penalty`) в соответствии с рекомендациями API. Дополнительно вводилось ограничение на длину генерируемого текста через параметр `max_tokens`, чтобы приблизить длину синтетических текстов к распределению длин комментариев в корпусе естественных текстов (средняя длина – 103 символа, таблица 1). Это

позволило сделать тексты более сопоставимыми по структуре и объему.

Каждому сгенерированному тексту присваивалась метка, отражающая источник (конкретная LLM), однако в рамках бинарной классификации все такие тексты рассматривались как единый обобщенный класс «сгенерировано LLM». Информация о наборе LLM-текстов приведена в таблице 2.

Таблица 2. Набор LLM-текстов

Модель	Количество текстов
LLaMA (Яндекс) [29]	100000
Yandex GPT [30]	100000
Yandex GPT-lite [30]	100000
GPT-4o [23]	100000
GPT-3.5 [23]	100000
GigaChat Pro [31]	100000
DeepSeek [32]	100000
Итого	700000

Также методика была протестирована на наборе данных Г. Грицай и др. [20] и датасете RuATD-2022 [21]. Информация о датасетах представлена в таблице 3.

Таблица 3. Характеристики датасетов Г. Грицай и RuATD-2022

Тип текста	Характеристика набора	Значение для датасета Г. Грицай	Значение для датасета RuATD-2022
Ест.	Кол-во текстов	451572	59674
Иск.		451572	155436
Ест.	Кол-во символов	1359568468	15679465
Иск.		1684485630	33655209
Ест.	Ср. длина текста, символов	3011	263
Иск.		3730	217
Ест.	Мин. длина текста, символов	500	7
Иск.		302	6
Ест.	Макс. длина текста, символов	429800	2963
Иск.		15281	3560

3.2. Признаковое пространство. Чтобы выявить неявные различия между естественными и сгенерированными текстами, из каждого текста извлекался набор признаков разного типа:

1. Лингвистические признаки. Данная группа включает признаки, характеризующие стиль и структуру текста на уровне явных отличимых человеком лингвистических особенностей. В частности, рассчитывалась средняя длина предложения – предполагалось, что LLM могут продуцировать более формализованные или однотипные по длине предложения. Также рассчитывалось количество знаков препинания (общее число и среднее) для отражения сложности синтаксической структуры. Морфологические признаки включали частотные распределения частей речи. Также в данную категорию признаков входили метрики лексического разнообразия: доля редких и распространенных слов. Для этого использовался частотный словарь русского языка [33].

2. Статистические признаки. К этой категории относятся количественные меры энтропии и вероятностной непредсказуемости текста. Вычислялась энтропия символов и слов – как среднее значение по всему тексту. Энтропия для последовательности символов определялись стандартно через частоты появления каждого символа. Аналогично вычислялась энтропия распределения слов, рассматривая частоты слов. Низкая энтропия указывает на однообразие (повторяемость одних и тех же слов или букв), тогда как высокая – на вариативность. Предполагалось, что сгенерированные тексты могут обладать отличной от естественных текстов статистической структурой (например, некоторые модели могут иметь избыточную равномерность или, наоборот, неестественно низкое разнообразие, особенно на коротких отрывках). Также рассчитывалась перплексия. Для расчета данного признака была обучена простая статистическая языковая модель (5-граммная модель) на корпусе естественных русских текстов, используя KenLM [34]. Полученная модель рассчитывала вероятность последовательности символов, и на ее основе вычислялась перплексия каждого рассматриваемого текста. Естественные тексты должны быть «более предсказуемыми» (низкая перплексия) относительно статистики человеческого языка, тогда как LLM-тексты могут содержать менее характерные сочетания слов, повышающие перплексию.

Также рассчитывались частоты биграмм и триграмм для выявления их повторяемости в тексте. Высокое число повторяющихся последовательностей слов и символов может свидетельствовать о том, что модель склонна к использованию определенных фраз или шаблонов, тогда как у человека в таком коротком тексте обычно меньше точных повторов.

3. Семантические признаки. Для выявления неявных признаков использовались эмбединги текста, полученные с помощью современных трансформерных моделей. Применен RuBERT от DeepPavlov [35]. С помощью RuBERT каждый текст был преобразован в вектор фиксированной размерности (768-мерный). Модель RuBERT обучена на русском корпусе Википедии и новостей, поэтому ее скрытые состояния отражают контекстуальные зависимости слов на русском языке. LLM при генерации склонны включать лишнюю информацию или использовать более общие формулировки, что приводит к характерным отличиям в эмбедингах. Некоторые координаты этих векторов (или их линейные комбинации) оказываются информативными для классификации и отражают стилистические и контекстуальные особенности текстов. Используются эмбединги RuBERT (*hidden size* = 768). Далее этот вектор подается на вход BiLSTM.

Для блока статистических и лингвистических признаков использовались доля знаков препинания, частота использования эмодзи, средняя длина слова, коэффициент лексического разнообразия (TTR), частоты стоп-слов, количество предложений, распределение частей речи, частота заглавных букв, частоты повторяющихся символов. Семантические признаки формировались на основе 768-мерных эмбедингов RuBERT и косинусных расстояний между предложениями.

Веса признаков для LightGBM оценивались по показателю прироста информации (gain) – рисунок 2.



Рис. 2. Оценка важности признаков

Наибольший вклад внесли: доля знаков препинания (0,18), коэффициент лексического разнообразия (TTR) (0,15), частота эмодзи (0,12) и средняя длина слова (0,10). В Transformer-модели наибольшие веса пришлись на знаки препинания и редкие слова, что коррелировало с важностью признаков в LightGBM.

3.3. Архитектура. Для решения задачи классификации («Человек-LLM») предложена ансамблевая архитектура, объединяющая несколько моделей различного типа (рисунок 3).

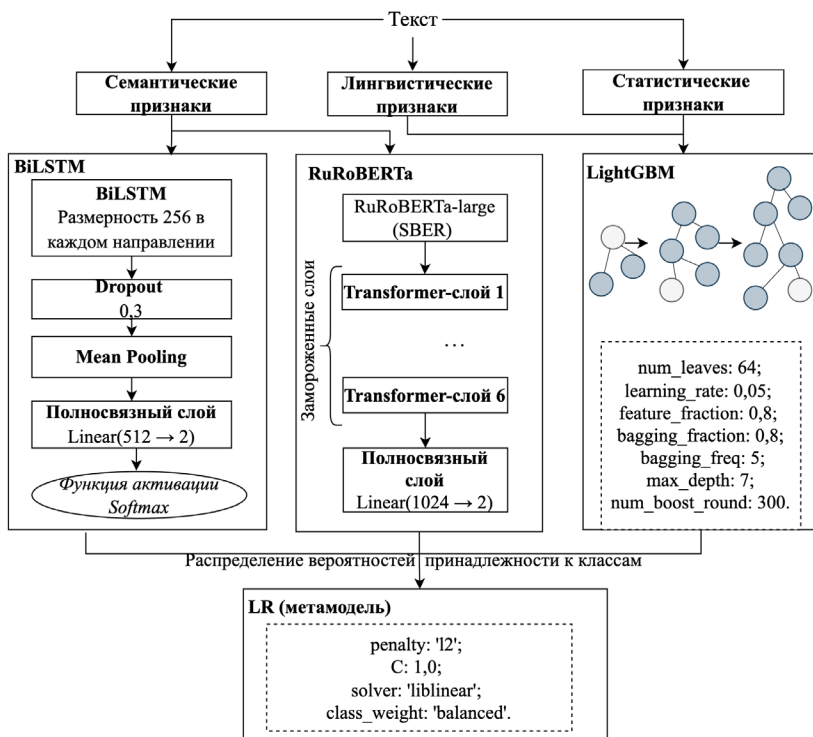


Рис. 3. Архитектура предложенной модели для выявления сгенерированных текстов

Такая комбинация позволяет учесть разнородные признаки текста (раздел 3.2). В состав ансамбля вошли следующие компоненты:

1. Модель градиентного бустинга LightGBM. LightGBM используется для обучения на лингвистических и статистических характеристиках, описанных в разделе 3.2. Вектор признаков каждого

текста подается на вход LightGBM. Модель выбрана из-за ее скорости обучения и способности справляться со смешанными по природе признаками, а также встроенной функции оценки важности признаков. Обученная модель выдает для каждого текста оценку принадлежности к классу «LLM-текст» (вероятность или логит). Параметры обучения LightGBM: objective: 'binary' (задача бинарной классификации (человек/LLM)); metric: 'binary_logloss' (логарифмическая функция потерь для бинарной классификации); boosting_type: 'gbdt' (градиентный бустинг по деревьям решений); num_leaves: 64 (максимальное количество листьев в одном дереве); learning_rate: 0,05 (шаг градиентного обновления); feature_fraction: 0,8 (доля признаков, случайно выбираемых для построения каждого дерева); bagging_fraction: 0,8; bagging_freq: 5 (обучение на случайных подвыборках каждые 5 итераций); max_depth: 7 (максимальная глубина дерева, ограничивающая переобучение); num_boost_round: 300 (общее количество итераций); early_stopping_rounds: 30 (остановка обучения при отсутствии улучшений).

2. BiLSTM. В качестве второго компонента выбрана модель двунаправленной рекуррентной нейросети с долгой краткосрочной памятью (LSTM) – BiLSTM. В данном случае на вход BiLSTM подается последовательность эмбеддингов, представляющих текст. К причинам выбора BiLSTM в качестве компонента ансамбля отнесем ее способность анализировать текст в двух направлениях, что особенно полезно для коротких текстов, где каждое слово особенно влияет на семантику всего текста; устойчивость к нестандартной грамматике и синтаксису – LLM часто генерируют тексты с гладкой, но шаблонной структурой. BiLSTM способна уловить тонкие различия в синтаксисе, которые могли бы ускользнуть от классических моделей.

Параметры модели BiLSTM: размерность входного эмбеддинга: 768; размер скрытого состояния: 256 (в каждом направлении, количество нейронов в каждом направлении рекуррентной сети); количество LSTM-слоев: 1; dropout: 0,3 (регуляризация, предотвращающая переобучение); агрегация выходов: усреднение (mean pooling); полносвязный классификационный слой: Linear(512 → 2) – полносвязный слой, объединяющий оба направления; активация: Softmax. Обучение производилось с использованием оптимизатора 'Adam', learning rate: 0,001 (скорость обучения), batch size: 32 (размер батча), число эпох: 10.

3. RuRoBERTa. Третий модуль ансамбля – предобученная модель-трансформер семейства RoBERTa, адаптированная под русский язык. В работе применена модель, предоставленная командой

SberDevices [36]. Для задачи бинарной классификации («Человек-LLM») к модели применялась техника fine-tuning, а именно, добавлен полносвязный классификационный слой, проведено дообучение модели методом градиентного спуска на размеченном датасете. Большие модели вроде RoBERTa способны обнаруживать тонкие стилистические различия, недоступные более простым моделям, поэтому данный компонент должен улавливать высокоуровневые признаки, специфичные для машинного текста. Модель дообучалась под задачу бинарной классификации «Человек-LLM» по следующей схеме: добавлен классификационный слой Linear(1024 $\rightarrow$ 2) – полносвязный слой; оптимизатор: 'AdamW'; learning rate: 2e-5 (небольшой шаг обучения, характерный для дообучения крупных моделей); batch size: 16 (размер батча); число эпох: 3; стратегия заморозки: первые 6 слоев трансформера заморожены; warmup steps: 500; scheduler: 'linear decay with warmup' (линейное уменьшение параметра learning rate после фазы warmup); weight decay: 0,01 (регуляризация весов); loss function: 'CrossEntropyLoss' (кросс-энтропия для задачи классификации); fine-tuning осуществлялся на размеченном корпусе с использованием библиотеки Hugging Face Transformers.

4. Метамодель (стекинг). Финальное решение принимает ансамблевый мета классификатор, который агрегирует выходы трех перечисленных моделей. На этапе обучения для каждого объекта вычислялись предсказанные вероятности базовых моделей. Затем эти три значения использовались как входные признаки для обучающейся метамодели. В качестве мета классификатора была использована логистическая регрессия (LR) чтобы не переобучиться на выходах. Такой подход соответствует технике стекинг – базовые алгоритмы обучаются отдельно, а затем их прогнозы комбинируются обученным способом. Параметры метамодели: penalty: 'l2' (L2-регуляризация для предотвращения переобучения); C: 1,0 (коэффициент регуляризации); solver: 'liblinear'; class_weight: 'balanced' (автоматическая корректировка весов классов для борьбы с дисбалансом). Настройка мета-классификатора осуществляется на основе обучающего корпуса коротких текстов (100–200 символов), что позволяет модели учитывать специфику признаков именно для этого диапазона длины. Динамическое изменение весов компонентов ансамбля в зависимости от длины текста в текущей версии не применяется, однако рассматривается как потенциальное улучшение при адаптации методики к более длинным образцам.

Выбор оптимизаторов «Adam» и «AdamW» обусловлен их стабильной сходимостью при работе с короткими текстами. Для снижения риска переобучения применялись регуляризация (dropout, weight decay), ранняя остановка (early stopping), а также подбор шага обучения. Альтернативные оптимизаторы (SGD, RMSProp) были протестированы на части выборки и показали более медленную сходимость и менее стабильные результаты в задаче бинарной классификации.

Выбор гибридного ансамблевого подхода обусловлен стремлением учесть признаки на разных уровнях иерархии текста и обеспечить надежность классификации в условиях разных тематик генерируемых текстов, различных версий и видов языковых моделей. Каждая из использованных моделей имеет свои сильные стороны, и их комбинация компенсирует слабости друг друга, а именно:

1. Лингвистические и статистические признаки дают понятные с точки зрения лингвистики индикаторы «нестественности» текста. Однако эти признаки могут быть не способны к выявлению сложных контекстуальных признаков и закономерностей. С другой стороны, эмбединги, как более глубокое признаковое представление, способны выявлять неявные семантические паттерны. Объединение через ансамбль позволяет учитывать и обрабатывать оба типа признаков.

2. В корпус включены тексты, созданные разными LLM, каждая из которых имеет свои характерные черты. Ансамбль лучше адаптируется к такому разнообразию источников текстов. Например, если новая модель генерации не употребляет редких слов (тем самым классифицируется одной из моделей как естественный текст), все еще могут сработать другие модели ансамбля. Таким образом, использование ансамбля, по сути, дает дополнительную защиту против разных видов отклонений LLM-текста, одни компоненты ансамбля реагируют на статистические аномалии, другие – на стилистические. Это особенно важно, поскольку LLM быстро развиваются, конкретная генеративная модель может утратить актуальность, если появится модель, генерирующая тексты в ином стиле. Представленная методика изначально более устойчива к появлению новых LLM: даже если они маскируют одни признаки, останутся задействованы другие.

3. Короткие тексты (в районе 100-200 символов) крайне сложны для классификации, т.к. признаковое пространство ограничено. Длинные тексты, наоборот, дают больше материала для анализа, но модели могут терять фокус на ключевых особенностях. Применение ансамбля является преимуществом в таких случаях,

поскольку LightGBM, опирающаяся на усредненные показатели, менее чувствительна к длине (метрика перплексии уже усреднена по всему тексту), тогда как BiLSTM и RoBERTa, способны выявлять локальные признаки LLM-текста, а не только глобальные. В случае коротких текстов ключевым фактором при принятии решения могут быть численные признаки (например, очень низкая энтропия символов сразу выдаст шаблонный сгенерированный ответ), а при анализе более длинных образцов – глубокие нейросетевые модели выявят несвойственный человеку ход рассуждения за счет выявления неявных локальных признаков через эмбединги.

Для свертки предсказаний разнородных моделей применяется логистическая регрессия, обучаемая на вероятностях, выдаваемых каждой из базовых моделей. Таким образом, веса (или степень доверия) к каждому классификатору в ансамбле не задаются вручную, а определяются в процессе обучения мета-классификатора, что позволяет учитывать специфику их работы на различных типах входных данных.

4. Результаты. Для надежной оценки качества классификатора и настройки гиперпараметров была применена процедура кросс-валидации на 5 фолдах. Итоговый корпус был сбалансирован и разделен на обучающую и тестовую выборки в соотношении 80:20, при этом тексты перемешаны случайным образом, чтобы ни одна конкретная тема или источник не преобладали только в одной из выборок.

Эксперименты проводились по двум задачам: бинарная классификация («Человек-LLM») и многоклассовая классификация (определение «Какой моделью создан текст?»). При проведении всех экспериментов наборы данных были сбалансированы, исключались тексты длиной менее 100 символов.

Результаты эксперимента представлены для задачи «Человек-LLM» в таблице 4, для задачи «Какой моделью создан текст?» – в таблице 5.

В дополнение к основным экспериментам мы провели *leave-one-model-out* тестирование, в котором на каждой итерации полностью исключали из обучения тексты одной LLM-семьи (например, Яндекс-модели или OpenAI) и тестировали только на ней. Аналогично, внутри выбранной LLM-семьи выполняли *leave-one-version-out* тест (например, обучали на GPT-3.5, тестировали на GPT-4o, и наоборот). Для каждой исключенной модели мы рассчитывали *Accuracy*, *Precision*, *Recall* и *F1*. Результаты данного эксперимента приведены в таблице 6.

Таблица 4. Результаты эксперимента «Человек-LLM»

Модель	Архитектура	Accuracy	Precision	Recall	F1
LLaMA (Яндекс)	RuRoBERTa	0,86±0,02	0,87±0,02	0,85±0,03	0,86±0,02
	BiLSTM	0,85±0,02	0,86±0,03	0,84±0,02	0,85±0,03
	LightGBM	0,82±0,03	0,83±0,02	0,81±0,02	0,82±0,02
	Ансамбль	0,9±0,03	0,91±0,02	0,89±0,03	0,9±0,03
Yandex GPT	RuRoBERTa	0,83±0,03	0,84±0,02	0,82±0,03	0,83±0,03
	BiLSTM	0,82±0,03	0,83±0,03	0,81±0,03	0,82±0,03
	LightGBM	0,78±0,03	0,79±0,03	0,77±0,02	0,78±0,03
	Ансамбль	0,87±0,03	0,88±0,03	0,86±0,02	0,87±0,03
Yandex GPT-lite	RuRoBERTa	0,83±0,03	0,84±0,02	0,82±0,03	0,83±0,03
	BiLSTM	0,83±0,02	0,84±0,03	0,82±0,02	0,83±0,02
	LightGBM	0,79±0,04	0,8±0,03	0,78±0,04	0,79±0,03
	Ансамбль	0,91±0,02	0,92±0,03	0,9±0,02	0,91±0,02
GPT-4o	RuRoBERTa	0,78±0,03	0,79±0,03	0,77±0,03	0,78±0,03
	BiLSTM	0,77±0,04	0,78±0,03	0,76±0,04	0,77±0,04
	LightGBM	0,75±0,03	0,76±0,03	0,74±0,03	0,75±0,03
	Ансамбль	0,82±0,02	0,83±0,03	0,81±0,02	0,82±0,02
GPT-3.5	RuRoBERTa	0,84±0,04	0,85±0,03	0,83±0,04	0,84±0,04
	BiLSTM	0,83±0,03	0,84±0,03	0,82±0,03	0,83±0,02
	LightGBM	0,78±0,03	0,79±0,03	0,77±0,03	0,78±0,03
	Ансамбль	0,88±0,04	0,89±0,03	0,87±0,04	0,88±0,03
GigaChat Pro	RuRoBERTa	0,9±0,04	0,91±0,04	0,89±0,04	0,9±0,03
	BiLSTM	0,89±0,03	0,9±0,03	0,88±0,02	0,89±0,03
	LightGBM	0,87±0,02	0,88±0,03	0,86±0,02	0,87±0,02
	Ансамбль	0,95±0,03	0,96±0,03	0,94±0,03	0,95±0,03
DeepSeek	RuRoBERTa	0,84±0,04	0,85±0,04	0,83±0,04	0,84±0,03
	BiLSTM	0,84±0,03	0,85±0,03	0,83±0,02	0,84±0,03
	LightGBM	0,8±0,02	0,81±0,02	0,79±0,02	0,8±0,03
	Ансамбль	0,89±0,03	0,9±0,03	0,88±0,03	0,89±0,03
Все модели	RuRoBERTa	0,8±0,02	0,81±0,02	0,79±0,02	0,8±0,02
	BiLSTM	0,81±0,03	0,82±0,03	0,8±0,03	0,81±0,03
	LightGBM	0,77±0,02	0,78±0,02	0,76±0,03	0,77±0,03
	Ансамбль	0,84±0,03	0,85±0,02	0,83±0,03	0,84±0,03

Таблица 5. Результаты эксперимента «Какой моделью создан текст?»

Модель	Accuracy	Precision	Recall	F1
Yandex модели (2 класса)	0,89±0,03	0,9±0,02	0,88±0,03	0,89±0,03
Open AI модели (2 класса)	0,76±0,02	0,77±0,03	0,75±0,02	0,76±0,02
Все модели (7 классов)	0,6±0,04	0,62±0,03	0,59±0,03	0,6±0,03

Таблица 6. Оценка обобщаемости методики при исключении моделей и семейств LLM-моделей

Исключенная модель	Accuracy	Precision	Recall	F1
Yandex GPT	0,8±0,04	0,79±0,03	0,8±0,03	0,8±0,03
Yandex GPT-lite	0,83±0,03	0,82±0,02	0,82±0,04	0,82±0,03
GigaChat Pro	0,9±0,04	0,88±0,03	0,89±0,03	0,89±0,03
DeepSeek	0,86±0,02	0,85±0,03	0,84±0,02	0,85±0,03
GPT-4o	0,76±0,04	0,74±0,03	0,75±0,03	0,75±0,02
GPT-3.5	0,83±0,04	0,83±0,02	0,81±0,02	0,82±0,02

Результаты подтверждают эффективность предложенной методики. В задаче «Человек-LLM» ансамбль продемонстрировал стабильную и высокую точность на различных LLM, включая модели, ориентированные на русский язык (Yandex GPT, GigaChat Pro) и мультилингвальные разработки (OpenAI, DeepSeek). Средняя точность распознавания для отдельных моделей достигала 0,95 (GigaChat Pro), при этом разброс по моделям составил 0,82–0,95 (таблица 5). Совокупная точность по всем LLM составила 0,84, что существенно превосходит другие решения, предназначенные для анализа русского языка за счет выбора моделей, входящих в состав ансамбля и комбинации различных видов признаков.

В задаче «Какой моделью создан текст» точность идентификации конкретной LLM достигала 0,89 для группы Yandex-моделей и 0,76 для моделей OpenAI (таблица 6). Несмотря на естественное снижение точности в эксперименте с 7 классами (до 0,6), результат остается приемлемым с учетом стилистического сходства между генераторами и их способности адаптировать стиль под естественный текст.

Эксперименты показали, что при тестировании на моделях, отсутствующих в обучении, качество снижается на 0,03–0,08. относительно классических сценариев эксперимента, но остается не менее 0,76.

Для сравнения предложенной методики с результатами других авторов было проведено два эксперимента на общедоступных наборах данных (датасет Г. Грицай и соавторов, RuATD-2022 [21]). Эксперимент (таблица 7) показывает, что методика способна демонстрировать высокую точность 0,89 на датасетах, содержащих тексты, отличающиеся от представленного в работе набора (по длине, стилистике содержимого, типам генеративных моделей и др.).

Таблица 7. Результаты экспериментов на общедоступных наборах данных

Датасет	Авторы	Метод	Результат
RuATD binary [21]	Г. Грицай и др. [38]	DeBERTa	F1-мера 0,76
	Команда МГУ [37]	Ансамбль моделей Transformer	Точность 0,83
	Авторы статьи	Ансамбль RuRoBERTa, LightGBM, BiLSTM, LR как метамодель	Точность 0,87
Набор данных Г. Грицай и др. [19]	Ю.В. Чехович и др. [38]	XLM-RoBERTa	Точность 0,89
	Авторы статьи	Ансамбль RuRoBERTa, LightGBM, BiLSTM, LR как метамодель	Точность 0,89

5. Выводы и обсуждение. Разработанная методика позволяет эффективно выявлять тексты, сгенерированные большими языковыми моделями. Комбинация лингвистических, статистических и семантических признаков, а также использование ансамбля классификаторов обеспечивают точность до 0,95. Полученные результаты подтверждают целесообразность применения данного подхода для задач модерации контента, анализа достоверности информации и противодействия дезинформации.

Среди ограничений методики можно отметить специфику разработки исключительно для анализа русскоязычных текстов, обучение на текстах разговорного стиля (комментариев пользователей социальных сетях).

Дальнейшие планы исследования включают проведение анализа и адаптацию методики к другим стилям текстов, включая научный, официально-деловой, публицистический и художественный, расширение существующего набора данных с появлением новых версий LLM. Также возможно совершенствование стеккиновой модели, в рамках которого добавляются дополнительные признаки (например, длина текста или степень перплексии), которые могут влиять на «доверие» к отдельным компонентам ансамбля.

Литература

1. Fedotova A., Romanov A., Kurtukova A., Shelupanov A. Digital authorship attribution in Russian-language fanfiction and classical literature // *Algorithms*. 2022. vol. 16. no. 1.
2. Романов А.С. Методология идентификации автора текста для решения задач информационной безопасности // *Вопросы кибербезопасности*. 2024. № 3(61). С. 120–128. DOI: 10.21681/2311-3456-2024-3-120-128.
3. Kurtukova A., Romanov A., Shelupanov A., Fedotova A. Complex cases of source code authorship identification using a hybrid deep neural network // *Future Internet*. 2022. vol. 14. no. 10. DOI: 10.3390/fi14100287.
4. Zellers R., Holtzman A., Rashkin H., Bisk Y., Farhadi A., Roesner F., Choi Y. Defending against neural fake news // *Proceedings of the 33rd Int. Conf. on Neural Information Processing Systems*. 2019. pp. 9054–9065.
5. Kuznetsov, K., Tulchinskii E., Kushnareva L., Magai G., Baranniko S., Nikolenko S., Piontkovskaya I. Robust AI-Generated Text Detection by Restricted Embeddings // *Findings of the Association for Computational Linguistics: EMNLP*. 2024. pp. 17036–17055. DOI: 10.18653/v1/2024.findings-emnlp.992.
6. Fraser K.C., Dawkins H., Kiritchenko S. Detecting AI-Generated Text: Factors Influencing Detectability with Current Methods // *Journal of Artificial Intelligence Research*. 2025. vol. 82. pp. 2233–2278. DOI: 10.1613/jair.1.16665.
7. Prajapati M., Baliarsingh S.K., Dora C., Bhoi A., Hota J., Mohanty J.P. Detection of AI-generated text using large language model // *2024 International Conference on Emerging Systems and Intelligent Computing (ESIC)*. 2024. pp. 735–740. DOI: 10.1109/ESIC60604.2024.10481602.
8. Mitchell E., Lee Y., Khazatsky A., Manning C.D., Finn C. Detectgpt: Zero-shot machine-generated text detection using probability curvature // *Proceedings of the 40th International Conference on Machine Learning (PMLR)*. 2023. pp. 24950–24962.
9. Lau H.T., Zubiaga A. Understanding the Effects of Human-written Paraphrases in LLM-generated Text Detection // *arXiv preprint arXiv:2411.03806*. 2024.
10. Wu J., Yang S., Zhan R., Yuan Y., Chao L.S., Wong D.F. A survey on LLM-generated text detection: Necessity, methods, and future directions // *Computational Linguistics*. 2025. pp. 275–338. DOI: 10.1162/coli_a_00549.
11. GPTZero. URL: gptzero.me (дата обращения: 15.05.2025).
12. Kaviani A., Pourhashem Kallehbasti M.M., Kazemi S., Firouzi E., Ghafari M. LLM security guard for code // *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. 2024. pp. 600–603. DOI: 10.1145/3661167.366126.
13. Wu L. Y., Segura-Bedmar I. AI-generated Text Detection with a GLTR-based Approach // *arXiv preprint. arXiv:2502.12064*. 2025.

14. OpenAI. URL: openai.com (дата обращения: 15.05.2025).
15. OriginalityAI. URL: <https://originality.ai/> (дата обращения: 15.05.2025).
16. AI Detector & Content Checker By Copyleaks. URL: <https://copyleaks.com/ai-content-detector> (дата обращения: 15.05.2025).
17. Writer. AI content detector. URL: <https://writer.com/ai-content-detector/> (дата обращения: 15.05.2025).
18. Tulchinskii E., et al. Intrinsic dimension estimation for robust detection of AI-generated texts // *Advances in Neural Information Processing Systems*. 2023. vol. 36. pp. 39257–39276.
19. Nikolaev K. Development of a Neural Network Model for Recognizing Russian-Language Generated Texts // *2024 IEEE Int. Multi-Conf. on Engineering, Computer and Information Sciences (SIBIRCON)*. IEEE, 2024. pp. 396–400. DOI: 10.1109/SIBIRCON63777.2024.10758447.
20. Gritsay G., Grabovoy A., Chekhovich Y. Open access dataset for machine-generated text detection in Russian. Mendeley Data. V2. 2023. DOI: 10.17632/4ynxfp3w53.2.
21. Shamardina T., et al. Findings of the ruatd shared task 2022 on artificial text detection in Russian // *arXiv preprint arXiv:2206.01583*. 2022.
22. RuATD. URL: <https://github.com/dialogue-evaluation/RuATD> (дата обращения: 15.05.2025).
23. Skrylnikov S., Posokhov P., Makhnytkina O. Artificial text detection in Russian language: A BERT-based approach // *Proc. Int. Conf. Dialogue*. 2022. pp. 1–7.
24. Gritsai G., Voznyuk A., Grabovoy A., Chekhovich Y. Are AI Detectors Good Enough? A Survey on Quality of Datasets With Machine-Generated Texts // *arXiv e-prints*. 2024. arXiv:2410.14677.
25. Pan L., et al. MarkLLM: An Open-Source Toolkit for LLM Watermarking // *Proc. of the 2024 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*. 2024. pp. 61–71. DOI: 10.18653/v1/2024.emnlp-demo.7.
26. Pham C.M., et al. TopicGPT: A Prompt-based Topic Modeling Framework // *arXiv e-prints*. 2023. arXiv:2311.01449.
27. Tong Z., Zhang H. A text mining research based on LDA topic modelling // *International conference on computer science, engineering and information technology*. 2016. pp. 201–210. DOI: 10.5121/csit.2016.60616.
28. Geroimenko V. Key Principles of Good Prompt Design // *The Essential Guide to Prompt Engineering: Key Principles, Techniques, Challenges, and Security Risks*. Cham: Springer Nature Switzerland. 2025. pp. 17–36.
29. Модели генерации текста. URL: <https://yandex.cloud/ru/docs/foundation-models/concepts/yandexgpt/models> (дата обращения: 15.05.2025).
30. Yandex GPT. URL: <https://ya.ru/ai/gpt> (дата обращения: 15.05.2025).
31. GiGaChat. URL: <https://giga.chat/> (дата обращения: 15.05.2025).
32. DeepSeek. URL: <https://www.deepseek.com/> (дата обращения: 15.05.2025).
33. Кузнецов С.А. Большой толковый словарь русского языка. Shangwu Yinshuguan, 2020. 1481 с.
34. KenLM. URL: <https://github.com/kpu/kenlm> (дата обращения: 15.05.2025).
35. Savkin M., Voznyuk A., Ignatov F., Korzanova A., Karpov D., Popov A., Konovalov V. DeepPavlov 1.0: Your Gateway to Advanced NLP Models Backed by Transformers and Transfer Learning // *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2024. pp. 465–474. DOI: 10.18653/v1/2024.emnlp-demo.47.
36. RuRoBERTa-large. URL: <https://huggingface.co/ai-forever/ruRoberta-large> (дата обращения: 15.05.2025).
37. Maloyan N., Nutfullin B., Ilyushin E. Dialog-22 ruatd generated text detection // *arXiv preprint arXiv:2206.08029*. 2022.

38. Gritsay G., Grabovoy A., Chekhovich Y. Automatic detection of machine generated texts: Need more tokens // 2022 Ivannikov Memorial Workshop (IVMEM). IEEE, 2022. pp. 20–26. DOI: 10.1109/IVMEM57067.2022.9983964.

Федотова Анастасия Михайловна — младший научный сотрудник, научно-инжиниринговый центр «интеллектуальные системы доверенного взаимодействия», ТУСУР. Область научных интересов: информационная безопасность, искусственный интеллект, машинное обучение, deep learning, NLP. Число научных публикаций — 20. afedotowaa@yandex.ru; проспект Ленина, 40, 634050, Томск, Россия; р.т.: +7(923)444-4125.

Романов Александр Сергеевич — канд. техн. наук, доцент кафедры, кафедра комплексной информационной безопасности электронно-вычислительных систем (кибэвс), ТУСУР. Область научных интересов: информационная безопасность, интеллектуальный анализ данных, искусственный интеллект, обработка текста. Число научных публикаций — 76. alexx.romanov@gmail.com; проспект Ленина, 40, 634050, Томск, Россия; р.т.: +7(923)444-4125.

Поддержка исследований. Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ в рамках базовой части государственного задания ТУСУРа на 2023–2025 гг. (проект № FEWM-2023-0015).

A. FEDOTOVA, A. ROMANOV
**TECHNIQUE FOR IDENTIFYING TEXTS GENERATED BY
LARGE LANGUAGE MODELS**

Fedotova A., Romanov A. Technique for Identifying Texts Generated by Large Language Models.

Abstract. The article presents a method for identifying Russian-language texts generated by large language models (LLMs). The method was developed with a focus on short messages from 100 to 200 characters long. The relevance of the work is due to the widespread use of generative models, such as GPT-3.5, GPT-4o, LLaMA, GigaChat, DeepSeek, and Yandex GPT. The method is based on an ensemble of machine learning models, and features of three levels are also used: linguistic (structure, punctuation, morphology, lexical diversity), statistical (entropy, perplexity, n-gram frequency), and semantic (RuBERT embeddings). LightGBM, BiLSTM, and the pre-trained transformer model RuRoBERTa are used as basic models, combined by stacking through logistic regression. The choice of a hybrid ensemble approach is due to the desire to take into account features at different levels of the text hierarchy and to ensure the reliability of classification in the context of different topics of generated texts, versions, and types of language models. The use of an ensemble is an advantage in the analysis of short texts, since LightGBM, based on averaged indicators, is less sensitive to length (the perplexity metric is already averaged over the entire text), while BiLSTM and RoBERTa are able to identify local features of an LLM text, and not just global ones. The dataset of natural texts includes more than 2.8 million user comments from the VK social network. The LLM text dataset contains 700 thousand texts generated by seven relevant large language models. Topic modeling (LDA) and role generation using prompt engineering were used in the text generation. The methodology was evaluated on open datasets of Russian-language LLM texts. The experimental results showed an accuracy of up to 0.95 in the binary classification task (Human–LLM) and up to 0.89 in the multi-class task of determining the model-generator. The method demonstrates robustness to the diversity of sources, styles, and LLM versions.

Keywords: large language models, neural networks, machine learning, text generation, ensemble of classifiers, text features.

References

1. Fedotova A., Romanov A., Kurtukova A., Shelupanov A. Digital authorship attribution in Russian-language fanfiction and classical literature. *Algorithms*. 2022. vol. 16. no. 1.
2. Romanov A.S. [Methodology for identifying the author of the text for solving information security problems]. *Voprosy kiberbezopasnosti – Cybersecurity issues*. 2024. no. 3(61). pp. 120–128. DOI: 10.21681/2311-3456-2024-3-120-128. (In Russ.).
3. Kurtukova A., Romanov A., Shelupanov A., Fedotova A. Complex cases of source code authorship identification using a hybrid deep neural network. *Future Internet*. 2022. vol. 14. no. 10. DOI: 10.3390/fi14100287.
4. Zellers R., Holtzman A., Rashkin H., Bisk Y., Farhadi A., Roesner F., Choi Y. Defending against neural fake news. *Proceedings of the 33rd Int. Conf. on Neural Information Processing Systems*. 2019. pp. 9054–9065.
5. Kuznetsov, K., Tulchinskii E., Kushnareva L., Magai G., Baranniko S., Nikolenko S., Piontkovskaya I. Robust AI-Generated Text Detection by Restricted Embeddings. *Findings of the Association for Computational Linguistics: EMNLP*. 2024. pp. 17036–17055. DOI: 10.18653/v1/2024.findings-emnlp.992.

6. Fraser K.C., Dawkins H., Kiritchenko S. Detecting AI-Generated Text: Factors Influencing Detectability with Current Methods. *Journal of Artificial Intelligence Research*. 2025. vol. 82. pp. 2233–2278. DOI: 10.1613/jair.1.16665.
7. Prajapati M., Baliarsingh S.K., Dora C., Bhoi A., Hota J., Mohanty J.P. Detection of AI-generated text using large language model. 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC). 2024. pp. 735–740. DOI: 10.1109/ESIC60604.2024.10481602.
8. Mitchell E., Lee Y., Khazatsky A., Manning C.D., Finn C. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *Proceedings of the 40th International Conference on Machine Learning (PMLR)*. 2023. pp. 24950–24962.
9. Lau H.T., Zubiaga A. Understanding the Effects of Human-written Paraphrases in LLM-generated Text Detection. *arXiv preprint arXiv:2411.03806*. 2024.
10. Wu J., Yang S., Zhan R., Yuan Y., Chao L.S., Wong D.F. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*. 2025. pp. 275–338. DOI: 10.1162/coli_a_00549.
11. GPTZero. Available at: gptzero.me (accessed 15.05.2025).
12. Kavian A., Pourhashem Kallehbasti M.M., Kazemi S., Firouzi E., Ghafari M. LLM security guard for code. *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. 2024. pp. 600–603. DOI: 10.1145/3661167.366126.
13. Wu L. Y., Segura-Bedmar I. AI-generated Text Detection with a GLTR-based Approach. *arXiv preprint. arXiv:2502.12064*. 2025.
14. OpenAI. Available at: openai.com (accessed 15.05.2025).
15. OriginalityAI. Available at: <https://originality.ai/> (accessed 15.05.2025).
16. AI Detector & Content Checker By Copyleaks. Available at: <https://copyleaks.com/ai-content-detector> (accessed 15.05.2025).
17. Writer. AI content detector. Available at: <https://writer.com/ai-content-detector/> (accessed 15.05.2025).
18. Tulchinskii E., et al. Intrinsic dimension estimation for robust detection of AI-generated texts. *Advances in Neural Information Processing Systems*. 2023. vol. 36. pp. 39257–39276.
19. Nikolaev K. Development of a Neural Network Model for Recognizing Russian-Language Generated Texts. 2024 IEEE Int. Multi-Conf. on Engineering, Computer and Information Sciences (SIBIRCON). IEEE, 2024. pp. 396–400. DOI: 10.1109/SIBIRCON63777.2024.10758447.
20. Gritsay G., Grabovoy A., Chekhovich Y. Open access dataset for machine-generated text detection in Russian. *Mendeley Data*. V2. 2023. DOI: 10.17632/4ynxfp3w53.2.
21. Shamardina T., et al. Findings of the the ruatd shared task 2022 on artificial text detection in Russian. *arXiv preprint arXiv:2206.01583*. 2022.
22. RuATD. Available at: <https://github.com/dialogue-evaluation/RuATD> (accessed 15.05.2025).
23. Skrylnikov S., Posokhov P., Makhnytkina O. Artificial text detection in Russian language: A BERT-based approach. *Proc. Int. Conf. Dialogue*. 2022. pp. 1–7.
24. Gritsai G., Voznyuk A., Grabovoy A., Chekhovich Y. Are AI Detectors Good Enough? A Survey on Quality of Datasets With Machine-Generated Texts. *arXiv e-prints*. 2024. *arXiv:2410.14677*.
25. Pan L., et al. MarkLLM: An Open-Source Toolkit for LLM Watermarking. *Proc. of the 2024 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*. 2024. pp. 61–71. DOI: 10.18653/v1/2024.emnlp-demo.7.
26. Pham C.M., et al. TopicGPT: A Prompt-based Topic Modeling Framework. *arXiv e-prints*. 2023. *arXiv:2311.01449*.

27. Tong Z., Zhang H. A text mining research based on LDA topic modelling. International conference on computer science, engineering and information technology. 2016. pp. 201–210. DOI : 10.5121/csit.2016.60616.
28. Geroimenko V. Key Principles of Good Prompt Design. The Essential Guide to Prompt Engineering: Key Principles, Techniques, Challenges, and Security Risks. Cham: Springer Nature Switzerland. 2025. pp. 17–36.
29. Модели генерации текста. Available at: <https://yandex.cloud/ru/docs/foundation-models/concepts/yandexgpt/models> (accessed 15.05.2025).
30. Yandex GPT. Available at: <https://ya.ru/ai/gpt> (accessed 15.05.2025).
31. GiGaChat. Available at: <https://giga.chat/> (accessed 15.05.2025).
32. DeepSeek. Available at: <https://www.deepseek.com/> (accessed 15.05.2025).
33. Kuznetsov S. A. Bol'shoj tolkovyj slovar' russkogo jazyka [Large explanatory dictionary of the Russian language]. Shangwu Yinshuguan, 2020. 1481 p. (In Russ.).
34. KenLM. Available at: <https://github.com/kpu/kenlm> (accessed 15.05.2025).
35. Savkin M., Voznyuk A., Ignatov F., Korzanova A., Karpov D., Popov A., Kononov V. DeepPavlov 1.0: Your Gateway to Advanced NLP Models Backed by Transformers and Transfer Learning. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2024. pp. 465–474. DOI: 10.18653/v1/2024.emnlp-demo.47.
36. RuRoBERTa-large. Available at: <https://huggingface.co/ai-forever/ruRoberta-large> (accessed 15.05.2025).
37. Maloyan N., Nutfullin B., Ilyushin E. Dialog-22 ruatd generated text detection. arXiv preprint arXiv:2206.08029. 2022.
38. Gritsay G., Grabovoy A., Chekhovich Y. Automatic detection of machine generated texts: Need more tokens. 2022 Ivannikov Memorial Workshop (IVMEM). IEEE, 2022. pp. 20–26. DOI: 10.1109/IVMEM57067.2022.9983964.

Fedotova Anastasiia — Junior researcher, Scientific and engineering center «intelligent systems of trusted interaction», TUSUR. Research interests: information security, artificial intelligence, machine learning, deep learning, NLP. The number of publications — 20. afedotowaa@yandex.ru; 40, Lenin Ave., 634050, Tomsk, Russia; office phone: +7(923)444-4125.

Romanov Aleksandr — Ph.D., Associate professor of the department, Department of integrated information security of electronic computing systems, TUSUR. Research interests: information security, data mining, artificial intelligence, text processing. The number of publications — 76. alexx.romanov@gmail.com; 40, Lenin Ave., 634050, Tomsk, Russia; office phone: +7(923)444-4125.

Acknowledgements. This research was funded by the Ministry of Science and Higher Education of Russia, Government Order for 2023–2025, project No. FEWM-2023-0015 (TUSUR).

А.А. БАЛЯБИН, С.А. ПЕТРЕНКО
**МОДЕЛЬ ОБЛАЧНОЙ ПЛАТФОРМЫ КРИТИЧЕСКОЙ
ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С
КИБЕРИММУНИТЕТОМ**

Балябин А.А., Петренко С.А. Модель облачной платформы критической информационной инфраструктуры с кибериммунитетом.

Аннотация. Исследование посвящено решению задачи синтеза модели облачной платформы критической информационной инфраструктуры с кибериммунитетом. Актуальность исследования обусловлена необходимостью разрешения проблемной ситуации, характеризующейся наличием противоречий в науке и практике. Противоречие в практике наблюдается между повышенными требованиями к устойчивости функционирования облачных платформ критической информационной инфраструктуры и ростом угроз, связанных с эксплуатацией новых, ранее неизvestных уязвимостей. Противоречие в науке состоит в невозможности обеспечения требуемой устойчивости таких платформ с использованием существующих моделей и методов. Так существующие подходы не в полной мере учитывают особенности облачных платформ критической информационной инфраструктуры, а именно, иерархическую архитектуру, наличие невыявленных уязвимостей, функционирование в условиях целенаправленных информационно-технических воздействий, повышенные требования к устойчивости и необходимость оперативного восстановления штатного функционирования. Поставлена задача синтеза новой модели облачной платформы критической информационной инфраструктуры с кибериммунитетом. Сформулирована гипотеза о том, что учет свойства кибериммунитета положительно влияет на устойчивость функционирования таких платформ в условиях информационно-технических воздействий. Методы исследования включают методы системного анализа, теории вероятностей, семантической теории программ, теории подобия и размерностей, а также методы компьютерной иммунологии. Обоснована идея кибериммунитета, состоящая в наделении облачной платформы способностью противодействовать известным и ранее неизвестным информационно-техническим воздействиям, оперативно восстанавливаться при возникновении нарушений и запоминать вредоносные входные данные, предотвращая их повторную обработку. Обоснованы показатели устойчивости функционирования облачных платформ критической информационной инфраструктуры. Разработана модель облачной платформы критической информационной инфраструктуры с кибериммунитетом. Научная новизна модели заключается в том, что в нее впервые введены такие элементы, как обнаружитель нарушений семантики вычислений, восстановитель штатного функционирования и кибериммунная память, в совокупности реализующие новое эмерджентное свойство кибериммунитета. Проведены теоретическое и экспериментальное исследования модели, по результатам которых подтверждена выдвинутая гипотеза. Практическая значимость результатов исследования заключается в доведении их до технических рекомендаций по архитектуре программного комплекса, которые могут быть использованы при разработке средств защиты облачных платформ критической информационной инфраструктуры, в частности, облачной платформы «ГосТех», в условиях информационно-технических воздействий.

Ключевые слова: облачные вычисления, семантика вычислений, критическая информационная инфраструктура, киберустойчивость, кибератаки, кибериммунитет.

1. Введение. В рамках реализации национального проекта «Экономика данных» в Российской Федерации ведется активная разработка и внедрение современных «сквозных» технологий, включая отечественные цифровые платформы и программное обеспечение, технологии искусственного интеллекта, блокчейн, квантовые вычисления и другие. Внедрение таких технологий необходимо для достижения цифровой трансформации государственного и муниципального управления, экономики и социальной сферы, обеспечения информационной безопасности и эффективного взаимодействия посредством сети Интернет.

Одним из примеров отечественных цифровых платформ является Единая цифровая платформа «ГосТех», позволяющая федеральным и региональным органам власти создавать государственные информационные системы (ГИС) и цифровые сервисы с использованием типовых программных решений, осуществлять мониторинг и управление ими. Платформа «ГосТех» может использоваться в качестве облачной платформы (ОП) для ГИС, относящихся к объектам критической информационной инфраструктуры (ОП КИИ).

В архитектуре типовых ОП можно условно выделить 4 уровня:

- системы виртуализации и гипервизоры;
- облачные операционные системы (ОС);
- промежуточные среды выполнения;
- прикладные облачные сервисы.

Следует отметить, что облачное программное обеспечение, функционирующее на различных уровнях ОП, характеризуется вложенной, иерархической структурой. В виртуальных средах (гипервизорах) могут быть параллельно запущены несколько изолированных ОС с контейнерными средами. Внутри контейнеров может выполняться множество микросервисов, таких как узел сети блокчейн, платформа машинного обучения, система управления базами данных, веб-сервер и другие.

Это говорит о наличии вертикальных (логических) связей между компонентами ОП. К типовым ОП относятся Microsoft Azure, Amazon Web Services, Google Cloud Platform и другие. Среди отечественных ОП известны, например, Yandex.Cloud, Selectel, MTC Web Services. Концептуальная модель такой ОП, учитывающая примерный стек технологий на каждом из уровней, представлена на рисунке 1.

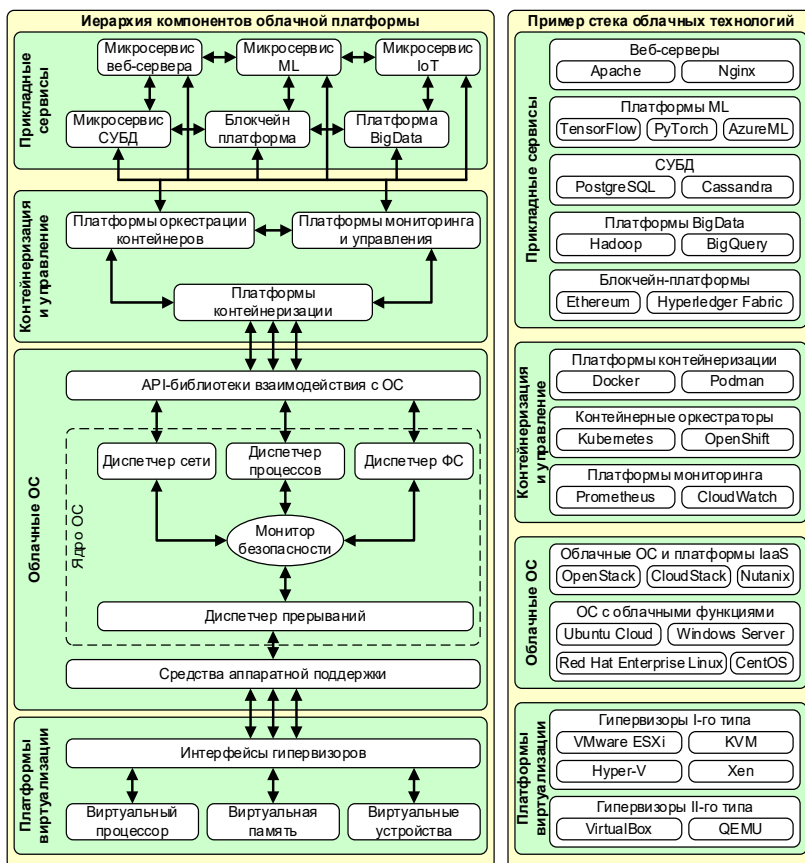


Рис. 1. Концептуальная модель ОП

Облачные платформы КИИ обладают некоторыми особенностями по отношению к типовым ОП.

Во-первых, в ОП КИИ может быть реализована мульти-облачная или гибридная архитектура, которая включает множество частных и публичных облачных сред. Такая архитектура предполагает наличие как вертикальных (логических), так и горизонтальных (информационных) связей между компонентами, что характеризует ОП КИИ как сложную иерархическую систему.

Во-вторых, ОП КИИ ориентированы на применение отечественных технологий и иных решений с открытым исходным кодом, однако в силу их ускоренной разработки и недостаточности

тестирования в них могут возникать новые, ранее неизвестные уязвимости. Это создает условия для реализации новых информационно-технических воздействий (ИТВ), направленных на нарушение семантики облачных вычислений (нарушение поведения программы, возникновение аномальных состояний при сохранении ее работоспособности), путем эксплуатации невыявленных уязвимостей.

В-третьих, информационные системы, создаваемые на базе ОП КИИ, предназначены для обеспечения деятельности органов власти. Успешная реализация ИТВ на такие платформы может привести к катастрофическим последствиям, поэтому к ним предъявляются повышенные требования в части безопасности и устойчивости их функционирования.

Таким образом, к значимым особенностям ОП КИИ относятся:

- сложная многоуровневая иерархическая архитектура;
- потенциальное наличие невыявленных уязвимостей;
- функционирование в условиях ИТВ, направленных на

нарушение семантики вычислений;

- повышенные требования к устойчивости и необходимость оперативного восстановления штатного функционирования.

В то же время в мире наблюдается устойчивая тенденция роста количества и сложности киберугроз. Воздействия приобретают целенаправленный, комплексный, многоэтапный характер и зачастую ориентированы на новые технологии, для которых еще в недостаточной степени разработаны методы и средства защиты. Так в исследовании CheckPoint [1] отмечается, что в начале 2025 года наблюдался рост эксплуатации уязвимостей облачных сред, связанных с их некорректной конфигурацией и уязвимостями публичных интерфейсов (API). По данным исследования, проведенного ГК «Солар» [2], в Российской Федерации в 2024 году наиболее значимыми угрозами для промышленности и телекома оказались АРТ-группировки и средства удаленного доступа. В целом же с 2022 года ежегодный прирост количества кибератак на информационную инфраструктуру России составляет порядка 20-30% [3].

Совокупность вышеуказанных факторов позволяет сформулировать *противоречие в практике* между повышенными требованиями к безопасности и устойчивости функционирования ОП КИИ и ростом угроз, связанных в том числе с преодолением существующих средств защиты и эксплуатацией новых, ранее неизвестных уязвимостей.

2. Анализ предметной области. Существует ряд перспективных интеллектуальных и биоинспирированных подходов к

моделированию и управлению устойчивостью различных информационно-вычислительных систем в условиях ИТВ, а именно, подходы на основе:

- графовых моделей [4, 5];
- марковских цепей [6, 7];
- гироматов безопасности [8, 9];
- управляющих графов программ [10];
- нейронных сетей [11 – 17];
- искусственных иммунных систем и сетей [19 – 27];
- самовосстанавливающихся вычислений и

кибериммунитета [29 – 34].

Рассмотрим подробнее данные подходы и их применимость для моделирования ОП КИИ.

В работе [4] предлагается обеспечивать информационную безопасность и устойчивость киберфизических систем (КФС) на основе принципов гомеостаза. Для этого КФС моделируется при помощи графов. Устойчивость функционирования КФС оценивается на основе анализа структурных свойств моделирующих графов, таких как избыточность и спектр графа [5]. Противодействие ИТВ, снижающим устойчивость функционирования КФС, предлагается осуществлять путем синтеза сценариев переконфигурирования. Фрактальные графы учитывают иерархический характер ОП КИИ, однако не позволяют учитывать ранее неизвестные уязвимости, связанные с нарушением семантики облачных вычислений.

В работе [6] исследуются компьютерные атаки на программно-конфигурируемые сети, с использованием математического аппарата цепей Маркова. В работе [7] их устойчивость оценивается с использованием вероятностно-временных показателей реализации кибератак. Цепи Маркова позволяют учесть стохастический характер возникновения нарушений, однако, моделирование сложных иерархических информационно-вычислительных систем в условиях ИТВ, направленных на нарушение семантики вычислений, может быть затруднительным.

В работах [8, 9] предлагается наделять интеллектуальные системы кибербезопасности свойством антиципации, позволяющим противодействовать кибератакам путем синтеза упреждающего поведения. Моделирование таких систем предлагается осуществлять на основе самообучающейся системы самоорганизующихся гироматов. Такая модель позволяет учесть многоуровневый иерархический характер ОП КИИ, наличие восстанавливающих воздействий, накопление информации о нарушениях, однако, является

«высокоуровневой» и не позволяет описывать ИТВ, направленные на нарушение семантики вычислений.

В работе [10] предлагается исследовать программное обеспечение на предмет наличия дефектов с использованием универсальных графовых представлений кода, учитывающих взаимодействия между компонентами. Управляющие графы позволяют описать иерархию взаимодействия вычислительных программ, однако, их выразительных свойств достаточно лишь для описания синтаксических структур. Для исследования нарушений семантики вычислений необходимо использовать дополнительные модели и методы.

Известны исследования, посвященные применению нейронных сетей и машинного обучения для задач выявления атак на IoT-системы [11], системы Smart Grid [12], киберфизические системы [13], а также для задач совершенствования систем обнаружения вторжений на основе классических алгоритмов машинного обучения [14], федеративного обучения [15], глубокого обучения [16], автокодировщика [17]. Нейросетевая модель позволяет выявлять как известные, так и некоторые ранее неизвестные ИТВ, однако, она не позволяет осуществлять восстановление штатного функционирования системы. Кроме того, системы обнаружения вторжений на основе алгоритмов машинного обучения и искусственного интеллекта могут быть уязвимы к различным атакам, направленным на искажение данных обучающей выборки [18].

В работе [19] предлагается применить искусственную иммунную систему для обнаружения вторжений. В работе [20] искусственные иммунные системы предлагается применять для создания интеллектуальной системы мониторинга безопасности промышленного интернета вещей. В работе [21] предлагается подход на основе интеллектуальных методов обнаружения сетевых атак с использованием нейронных, нейронечетких и иммунных детекторов. Подходы, описанные в исследованиях [22 – 24], основаны на адаптации алгоритмов функционирования иммунных систем и иммунного ответа для противодействия компьютерным атакам. К этому же направлению относятся подходы на основе алгоритмов дендритных клеток [25], отрицательного отбора [26], клональной селекции [27]. Модели, предложенные в данных исследованиях, применимы для распределенных многоагентных систем и позволяют обнаруживать как известные, так и ранее неизвестные ИТВ, однако не учитывают семантику вычислений и не позволяют осуществлять восстановление штатного функционирования систем.

В целом в работе [28] отмечается необходимость количественного оценивания устойчивости функционирования объектов КИИ в условиях воздействия угроз нарушения информационной безопасности и невозможность осуществления такой оценки существующими методами без ряда допущений.

Результаты анализа позволяют сформулировать *противоречие в науке* между повышенными требованиями к безопасности и устойчивости функционирования ОП КИИ и невозможностью их обеспечения с использованием существующих моделей и методов.

Выявленные противоречия в науке и практике характеризуют проблемную ситуацию, разрешение которой является *актуальной* научной задачей. Перспективной для разрешения данной проблемной ситуации является идея организации самовосстанавливающихся вычислений и кибериммунитета. Методологические основы данного научного направления сформированы в монографиях [30, 31].

Настоящее исследование продолжает и развивает исследования [32 – 34], в которых определены основные угрозы безопасности и устойчивости функционирования ОП КИИ, и посвящено разработке модели ОП КИИ с кибериммунитетом в условиях ИТВ, направленных на нарушение семантики облачных вычислений.

3. Постановка задачи исследования. Выполним постановку задачи синтеза модели ОП КИИ с кибериммунитетом.

Дано:

L – ОП КИИ, функционирующая на N уровнях, так что $L = \{L_i \mid i \in [1, N]\}$, где L_i – подсистема i -го уровня;

$X = \{X_i \mid i \in [1, N]\}$ – множество входных данных ОП КИИ, где X_i – входные данные на i -ом уровне;

$Y = \{Y_i \mid i \in [1, N]\}$ – множество выходных данных ОП КИИ, где Y_i – выходные данные на i -ом уровне;

$A = \{A_i \mid i \in [1, N]\}$ – множество параметров вредоносных воздействий на ОП КИИ, где A_i – параметры вредоносных воздействий i -ом уровне;

$D = \{D_i \mid i \in [1, N]\}$ – множество параметров нейтрализующих воздействий на ОП КИИ, где D_i – нейтрализующие воздействия на i -ом уровне;

$f = \{f_i \mid i \in [1, N]\}$ – множество вычислительных программ ОП КИИ, где f_i – вычислительная программа на i -ом уровне;

$C = \{C_i \mid i \in [1, N]\}$ – множество подсистем кибериммунитета ОП КИИ, где C_i – подсистема кибериммунитета на i -ом уровне;

$K = \{K_i \mid i \in [1, N]\}$ – множество подсистем управления устойчивостью функционирования ОП КИИ, где K_i – подсистема управления устойчивостью функционирования на i -ом уровне;

$R = \{R_i \mid i \in [1, N]\}$ – множество показателей устойчивости функционирования ОП КИИ, где R_i – показатели устойчивости функционирования на i -ом уровне.

Под *кибериммунитетом* в настоящей работе будем понимать свойство ОП КИИ, заключающееся в способности противодействовать ИТВ, не допуская нарушений, оперативно восстанавливать машинные вычисления в случае возникновения таких нарушений, и обеспечивать требуемые значения показателей устойчивости функционирования.

Необходимо: разработать математическую модель M ОП КИИ L с кибериммунитетом, устанавливающую закономерность изменения множества выходных данных Y и множества показателей устойчивости ее функционирования R от множества входных данных X , множества значений параметров дестабилизирующих воздействий A и множества параметров нейтрализующих воздействий D , с учетом наличия множества подсистем кибериммунитета C и множества подсистем оценивания и управления устойчивостью R . При этом на значения X , Y , A , D наложены условия допустимости: $X \subseteq X_{\text{доп}}$, $Y \subseteq Y_{\text{доп}}$, $A \subseteq A_{\text{доп}}$, $D \subseteq D_{\text{доп}}$.

Формальная постановка задачи синтеза модели: найти

$$\begin{aligned} M &: L, X, Y, A, D, f, C, K \rightarrow Y, R \\ X &\subseteq X_{\text{доп}}, Y \subseteq Y_{\text{доп}}, A \subseteq A_{\text{доп}}, D \subseteq D_{\text{доп}}. \end{aligned}$$

Гипотеза исследования. Учет свойства кибериммунитета положительно влияет на устойчивость функционирования ОП КИИ в условиях ИТВ.

4. Обоснование идеи кибериммунитета. Иерархию вычислительных программ ОП КИИ можно представить в виде универсальной машины Тьюринга (МТ) большой вложенности:

$$T_{N_i} : \left(T_{N_{i-1}} \times \left(\dots \times \left(T_{N_0} \times X_{N_0}^{n_0} \right)^{n_1} \dots \right)^{n_{i-1}} \right)^{n_i} \rightarrow Y_{N_i},$$

где T_{N_i} – универсальная МТ N_i -го уровня; $X_{N_i}^{n_i}$ – набор из n_i входных данных для универсальной МТ T_{N_i} ; Y_{N_i} – выходные данные универсальной МТ T_{N_i} .

Каждая такая МТ реализует некоторую вычислимую функцию над строками $f: X \rightarrow Y$ так, что $\forall x \in X, \exists y \in Y: f(x) = y$ где X – множество входных данных (строк); Y – множество выходных данных (результатов преобразования строк). При этом МТ одного уровня могут применяться последовательно, формируя композиции функций $f_1 \circ f_2 \circ \dots \circ f_n$, которые без потери общности могут быть заменены составной функцией f .

Пусть T_1 – МТ, соответствующая спецификации программы и задающая функцию $f: X \rightarrow Y$, а T_2 – МТ, соответствующая фактической ее реализации и задающая функцию $g: X \rightarrow Y$. В работах [32, 33] показано, что если программа, реализуемая машиной T_2 , содержит семантическую ошибку, то множество ее входных данных подразделяется на два непересекающихся подмножества $X = X^+ \cup X^-$, $X^- \neq \emptyset$, где $X^+ = \{x \mid x \in X, f(x) = g(x)\}$ – множество строк, одинаково преобразуемых обеими МТ, $X^- = \{x \mid x \in X, f(x) \neq g(x)\}$ – множество строк, результаты преобразования которых различаются. Это является причиной возникновения новых, ранее неизвестных уязвимостей, сохраняющих синтаксическую корректность программы, но нарушающих семантику вычислений (смысл выполняемых вычислительных операций). Тогда реализация новых ИТВ становится возможной при эксплуатации таких уязвимостей атакующим путем нахождения строк, принадлежащих множеству X^- , и передачи их программе в качестве входных данных.

Любой алгоритм, реализуемый МТ, может быть представлен в виде вычислительной программы, которая, в свою очередь, имеет структуру управления и может быть описана графом $\Gamma(B, T)$, где $B = \{b_i \mid i \in [1, k]\}$ – множество линейных блоков, содержащих последовательности вычислительных операций; k – количество линейных блоков; $T = \{B \times B\}$ – множество связей по управлению между линейными блоками.

Для обеспечения возможности контроля семантики вычислений необходимо осуществить гомоморфное отображение управляющего

графа программы $\Gamma(B, T)$ в эталонную модель $M(S, A)$, представляющую собой граф, вершины которого содержат инварианты, описывающие допустимые состояния программы в линейных блоках в терминах формальных семантик и не зависящие от конкретной реализации программы, с сохранением переходов между ними. На рисунке 2 представлен пример семантической модели программы и выявляемых с помощью нее нарушений.

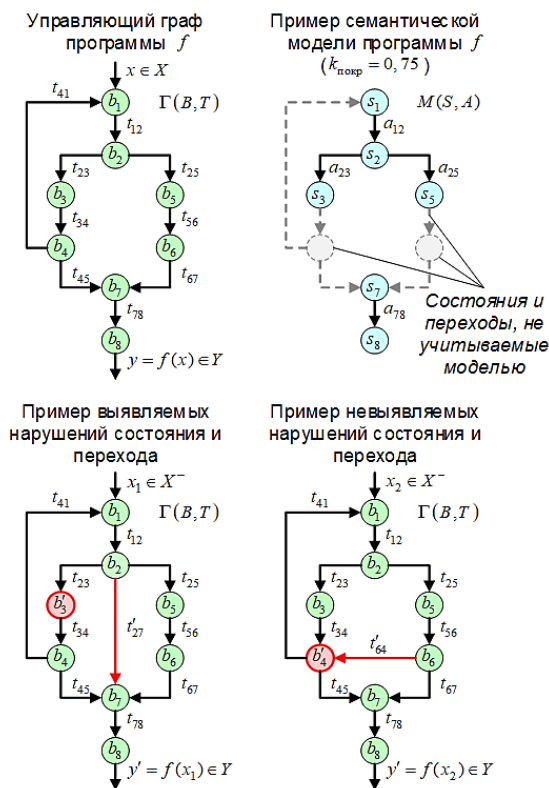


Рис. 2. Примеры управляющего графа, семантической модели программы, выявляемых и невыявляемых нарушений

Для контроля корректности семантики облачных вычислений в процессе функционирования ОП КИИ в условиях ИТВ потребуется внедрить в вычислительные программы дополнительную структурно-функциональную избыточность. Это может привести к недопустимому

увеличению времени их выполнения. Для решения данной проблемы может быть введен коэффициент покрытия кибериммунитета $k_{\text{покр}}$, определяющий долю линейных блоков, подлежащих контролю (на рисунке 2 $k_{\text{покр}} = 0,75$), однако неполное покрытие программы механизмами кибериммунитета приведет к пропуску некоторых нарушений, каждое из которых может быть расценено как потенциально критическое (приводящее к полному отказу системы или возникновению необратимых последствий). Для исследования влияния $k_{\text{покр}}$ на устойчивость функционирования и время выполнения программ ОП КИИ введем ряд показателей.

5. Обоснование показателей устойчивости. Примем некоторые допущения относительно функционирования ОП КИИ [29].

1. Программы ОП КИИ (сетевой драйвер, система управления базами данных, веб-сервер) осуществляют обработку входных данных последовательно, в порядке их поступления, и функционируют циклически так, что после завершения цикла обработки они переходят к ожиданию следующих входных данных. Таким образом, невозможно возникновение нескольких нарушений в единицу времени.

2. Нарушения обусловлены внешними факторами и не зависят от внутренних параметров ОП КИИ. Кроме того, для вредоносных входных данных, поступивших на предыдущих итерациях функционирования ОП КИИ, отсутствует эффект последствия. То есть, ранее произошедшие нарушения, не оказывают влияния на текущую итерацию функционирования ОП КИИ. При этом вредоносными являются только те входные данные, обработка которых непосредственно привела к нарушению семантики вычислений.

3. Образцы вредоносных входных данных сохраняются в кибериммунной памяти, а вновь поступающие входные данные принимаются только при отсутствии их среди сохраненных вредоносных образцов. После каждого обнаружения случайного или преднамеренного нарушения осуществляется восстановление штатного функционирования ОП КИИ. При этом, входные данные, переданные администратором и не являющиеся вредоносными, но приведшие к нарушению, также будут классифицированы как вредоносные (нарушающие функционирование ОП КИИ). С точки зрения управления безопасностью, это потребует дальнейшего анализа и устранения причин такого нарушения.

4. ИТВ характеризуются массовой отправкой нарушителем вредоносных входных данных, которые вместе с остальными

входными данными последовательно обрабатываются в порядке их поступления. Новые, ранее неизвестные ИТВ связаны с отправкой нарушителем новых вредоносных входных данных.

Введем теперь показатели устойчивости функционирования ОП КИИ. Одним из таких показателей является вероятность успешного противодействия ИТВ $P_{\text{прот}}$, которую можно выразить через обратную вероятность нарушения $P_{\text{прот}} = 1 - P_{\text{нар}}$.

Среднее время между пропусками атак $T_{\text{ср.атак}}$ характеризует способность ОП КИИ противодействовать ИТВ и является показателем средней продолжительности ее функционирования между возникновением нарушений. Данный показатель можно определить как отношение времени вычислений программы с учетом проверок наличия нарушений за $n_{\text{ц}}$ циклов к ожидаемому количеству нарушений за это время:

$$T_{\text{ср.атак}} = \frac{n_{\text{ц}}(T_{\text{выч}} + T_{\text{обн}})}{n_{\text{ц}}(1 - P_{\text{прот}})} = \frac{T_{\text{выч}} + T_{\text{обн}}}{1 - P_{\text{прот}}}. \quad (1)$$

Вероятность успешного восстановления $P_{\text{восст}}$ характеризует способность ОП КИИ оперативно восстанавливать штатное функционирование после обнаружения нарушения. Предполагается, что всякое обнаруженное нарушение семантического состояния или перехода может быть нейтрализовано, а значит $P_{\text{восст}} = P_{\text{обн}}$, где $P_{\text{обн}}$ определяется типом применяемой формальной семантики и будет обоснована позднее.

Среднее время восстановления $T_{\text{ср.восст}}$ характеризует оперативность восстановления функционирования ОП КИИ после нарушения. Данный показатель можно вычислить как отношение времени, затрачиваемого на восстановление функционирования ОП КИИ за $n_{\text{ц}}$ циклов, к ожидаемому числу восстановлений за это время:

$$T_{\text{ср.восст}} = \frac{n_{\text{ц}} T_{\text{восст}}}{n_{\text{ц}} P_{\text{восст}}} = \frac{T_{\text{восст}}}{P_{\text{восст}}}. \quad (2)$$

Вероятность работоспособности ОП КИИ $P_{\text{раб}}$ является показателем того, в данный момент ОП КИИ не находится в состоянии

нарушения. Это означает, что ОП КИИ противодействует ИТВ, сохраняя работоспособное и корректное состояние, или к данному моменту возникшее ранее нарушение было нейтрализовано. Данный показатель схож по смыслу с коэффициентом готовности и может быть вычислен как:

$$P_{\text{раб}} = \frac{T_{\text{ср.атак}}}{T_{\text{ср.атак}} + T_{\text{ср.восст}}} . \quad (3)$$

Вероятность достижения цели $P_{\text{цели}}$ характеризует интегральную способность ОП КИИ противодействовать ИТВ, не допуская нарушений семантики, оперативно восстанавливать машинные вычисления при их возникновении и сохранять устойчивое состояние, в котором возможно достижение цели функционирования ОП КИИ. Данный показатель можно вычислить как:

$$P_{\text{цели}} = (P_{\text{прот}} + (1 - P_{\text{прот}})P_{\text{восст}})P_{\text{раб}} . \quad (4)$$

Полное время выполнения программного цикла $T_{\text{вып}}$, учитывающее время непосредственно вычислений, динамического формирования семантических инвариантов и сравнение их с эталонными для обнаружения нарушений семантики облачных вычислений, а также время, затрачиваемое на восстановление штатного функционирования программ ОП КИИ, можно вычислить как:

$$T_{\text{вып}} = T_{\text{выч}} + T_{\text{обн}} + T_{\text{восст}} . \quad (5)$$

Основными оцениваемыми показателями устойчивости функционирования ОП КИИ в условиях ИТВ можно считать вероятность достижения цели $P_{\text{цели}}$ и время выполнения программного цикла $T_{\text{вып}}$.

Формализуем теперь модель ОП КИИ с учетом принятых допущений и введенных показателей устойчивости.

6. Формализация модели облачной платформы критической информационной инфраструктуры с кибериммунитетом. Введем ряд обозначений для ОП КИИ на i -ом уровне [29].

$X_i = \{X_{ij} \mid j \in [1, n_i]\}$ – множество входных данных (строк), принимаемых программами ОП КИИ i -го уровня, n_i – количество вычислительных программ i -го уровня;

$Y_i = \{Y_{ij} \mid j \in [1, n_i]\}$ – множество выходных данных программ ОП КИИ i -го уровня;

$k_{извi}$ – доля известных ИТВ в общем количестве ИТВ на i -ом уровне, $k_{извi} \in [0, 1]$. Оценивание данного параметра может осуществляться с помощью экспертных методов или в результате учений с имитацией действий атакующих;

$k_{вредi}$ – доля вредоносных входных данных в общем количестве входных данных i -го уровня, $k_{вредi} \in [0, 1]$. Оценивание данного параметра может также осуществляться с помощью экспертных методов или в результате учений с имитацией действий атакующих;

$A_i = \{k_{извi}, k_{вредi}\}$ – множество параметров ИТВ на i -ом уровне;

$k_{покрi}$ – коэффициент покрытия кибериммунитета i -го уровня, определяющий долю линейных блоков программы, для которых осуществляется контроль семантики, $k_{покрi} \in [0, 1]$. Данный коэффициент является основным варьируемым параметром, используемым при оценивании устойчивости ОП КИИ;

$D_i = \{k_{покрi}\}$ – множество параметров нейтрализующих воздействий на i -ом уровне;

M_i – эталонная семантическая модель программы i -го уровня;

k_i – среднее количество линейных блоков программ i -го уровня, $k_i \in \mathbb{N}$;

m_i – среднее количество инструкций в каждом линейном блоке программ i -го уровня, $m_i \in \mathbb{N}$;

n_i – среднее количество параметров в каждой инструкции i -го уровня, $n_i \in \mathbb{N}$;

$f_i = \{M_i, k_i, m_i, n_i\}$ – вычислительная программа i -го уровня;

I_i – кибериммунная память i -го уровня;

O_i – обнаружитель нарушений i -го уровня;

V_i – восстановитель штатного функционирования i -го уровня;

$C_i = \{I_i, O_i, V_i\}$ – подсистема кибериммунитета i -го уровня;

K_i – подсистема оценивания и управления устойчивостью i -го уровня;

$P_{\text{мод}i}$ – вероятность возникновения модификации вычислений при обработке вредоносных входных данных на i -ом уровне. Данная вероятность определяется на основе информации о вероятностях появления различных типов модификаций вычислительных структур;

$P_{\text{обн}i}$ – вероятность обнаружения нарушений, вызванных обработкой вредоносных входных данных на i -ом уровне. Данная вероятность определяется с использованием оценки коэффициента $k_{\text{изв}i}$;

$P_{\text{прот}i}$ – вероятность успешного противодействия ИТВ на i -ом уровне. Данная вероятность определяется с использованием известных параметров моделирования и оценок коэффициентов $k_{\text{изв}i}$ и $k_{\text{вред}i}$;

$T_{\text{ср.атак}i}$ – среднее время между пропусками ИТВ (атак) на i -ом уровне, $T_{\text{ср.атак}i} \in \mathbb{N}$ (ед. времени);

$P_{\text{восст}i}$ – вероятность успешного восстановления после нарушения на i -ом уровне. Данная вероятность определяется на основе значения $P_{\text{обн}i}$;

$T_{\text{ср.восст}i}$ – среднее время, затрачиваемое на восстановление после ИТВ на i -ом уровне, $T_{\text{ср.восст}i} \in \mathbb{N}$ (ед. времени);

$P_{\text{раб}i}$ – вероятность нахождения ОП КИИ на i -ом уровне в работоспособном состоянии в произвольный момент времени. Данная вероятность определяется с использованием известных параметров моделирования и оценок коэффициентов $k_{\text{изв}i}$ и $k_{\text{вред}i}$;

$P_{\text{цели}i}$ – вероятность достижения цели функционирования ОП КИИ в условиях ИТВ на i -ом уровне. Данная вероятность определяется с использованием известных параметров моделирования и оценок коэффициентов $k_{\text{изв}i}$ и $k_{\text{вред}i}$;

$T_{\text{вып}i}$ – время выполнения программного цикла, включая время вычислений, время обнаружения и время восстановления нарушений на i -ом уровне, $T_{\text{вып}i} \in \mathbb{N}$ (ед. времени);

$R_i = \{P_{\text{проти}}, T_{\text{ср.атаки}}, P_{\text{восст}}, T_{\text{ср.восст}}, P_{\text{раб}}, P_{\text{цели}}, T_{\text{вып}}\}$ – множество показателей устойчивости функционирования ОП КИИ i -го уровня;

$T_{\text{выч}i}$ – время вычислений программы i -го уровня, $T_{\text{выч}i} \in \mathbb{N}$ (ед. времени);

$T_{\text{обн}i}$ – время проверки вычислений на предмет наличия нарушений на i -ом уровне (время обнаружения нарушений), $T_{\text{обн}i} \in \mathbb{N}$ (ед. времени);

$T_{\text{восст}i}$ – время восстановления штатного функционирования программы i -го уровня, $T_{\text{восст}i} \in \mathbb{N}$ (ед. времени);

t_0 – время выполнения элементарной операции (при расчетах принимается $t_0 = 1$ ед. времени).

Обоснование аналитических выражений вероятностей $P_{\text{проти}}$, $P_{\text{мод}i}$, $P_{\text{обн}i}$, $P_{\text{цели}}$ в работе будет дано позднее.

Общая схема предлагаемой модели ОП КИИ с кибериммунитетом представлена на рисунке 3.

Рассмотрим подробнее подсистему кибериммунитета ОП КИИ.

Подсистема кибериммунитета ОП КИИ C_i предназначена для обнаружения нарушений семантики облачных вычислений, возникающих при обработке вредоносных входных данных $x \in X_i^- \subset X_i$, оперативного восстановления машинных вычислений, а также сохранения вредоносных образцов для предотвращения повторной обработки вредоносных входных данных и превентивного реагирования на схожие ИТВ в будущем.

Для наделения программ свойством кибериммунитета и обеспечения возможности обнаружения нарушений, как упоминалось ранее, в программу вносится избыточность, а также формируется ее эталонная семантическая модель. Линейные блоки программы могут быть представлены в виде последовательностей арифметических выражений. Тогда для формирования эталонной семантической модели можно использовать теорию подобия и размерностей [35]. Соотношения размерностей сохраняют свойства операндов арифметических выражений в динамике выполнения программы и служат инвариантами ее семантических состояний в линейных блоках.

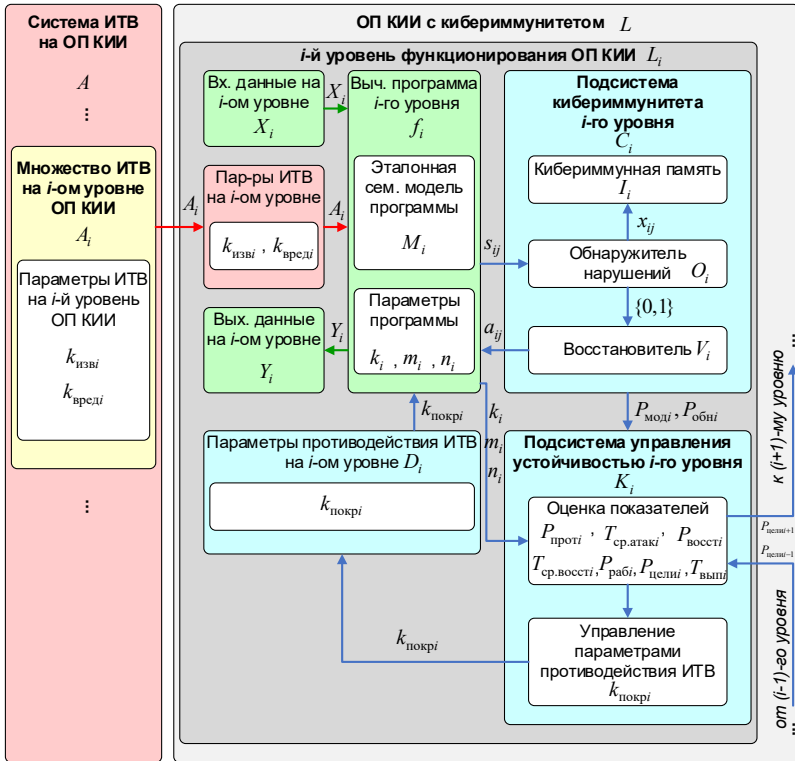


Рис. 3. Модель ОП КИИ с кибериммунитетом

Обозначим гомоморфное отображение управляющего графа программы в эталонную семантическую модель как $\mu: B, T \rightarrow S, A$, где множество линейных блоков программы B отображается в множество семантических состояний S , а множество переходов между блоками T отображается в множество переходов между семантическими состояниями A . По завершении преобразования в линейные блоки $\forall b_j \in B, b_j = (b_{j1}, \dots, b_{jm_i})$ вносятся метки для последующего контроля семантики вычислений так, что $b_j = (KT_j^{in}, b_{j1}, \dots, b_{jm_i}, KT_j^{out})$, где KT_j^{in} и KT_j^{out} – метки начала и конца линейного блока соответственно.

Вычислительный процесс представляет собой последовательную смену семантических состояний под управлением

переходов $s_{i1} \xrightarrow{a_{12}} s_{i2} \xrightarrow{a_{23}} \dots \xrightarrow{a_{ij-1}} s_{ij}$. Обнаружитель нарушений O_i формирует семантический инвариант программы, описывающий ее состояние в текущем линейном блоке, и осуществляет проверку его принадлежности эталонной семантической модели программы. В функциональном виде это может быть записано как $\eta : \langle S, M \rangle \rightarrow \{0, 1\}$, при этом:

$$\eta(s_{ij}, M_i) = \begin{cases} 1, & \text{если } s_{ij} \in S_i \wedge \exists a_{ij} : s_{ij-1} \xrightarrow{a_{ij}} s_{ij} \\ 0, & \text{иначе} \end{cases}$$

При обнаружении семантического состояния s'_{ij} , не удовлетворяющего эталонной модели, восстановитель V_i , реализующий функцию $\xi : \langle S, S \rangle \rightarrow A$, осуществляет синтез перехода для приведения вычислительной программы в корректное состояние в зависимости от принятой стратегии восстановления: начальное, предыдущее или конечное, как показано на рисунке 4.

Последовательность семантических состояний вычислительного процесса

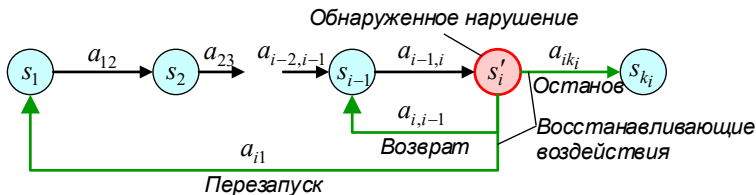


Рис. 4. Пример обнаружения и восстановления нарушения

Слова $x \in X_i^-$, которые привели к нарушению, накапливаются в памяти кибериммунитета I_i так, что $I_i \subseteq X_i^- \subset X_i$. В дальнейшем данные слова используются при проверке вновь поступающих входных данных для превентивного реагирования на аналогичные ИТВ. Таким образом происходит последовательное формирование приобретаемого кибериммунитета и самообучение ОП КИИ.

Таким образом, кибериммунитет является новым эмерджентным свойством, которое обеспечивается совокупностью

элементов модели ОП КИИ: обнаружителя O_i , восстановителя V_i и кибериммунной памяти I_i , с учетом коэффициента покрытия $k_{\text{покр}}$.

Рассмотрим теперь подсистему управления устойчивостью функционирования ОП КИИ.

Вероятность успешного противодействия ИТВ можно определить как $P_{\text{проти}} = 1 - P_{\text{нари}}$, где $P_{\text{нари}}$ – вероятность пропуска нарушения. Здесь вероятность пропуска нарушения можно определить как $P_{\text{нари}} = P_{S_3} (P_{S_1} + P_{S_2})$, где P_{S_1} – вероятность того, что нарушение произошло в контролируемом участке управляющего графа, но не было обнаружено; P_{S_2} – вероятность того, что нарушение произошло за пределами контролируемых участков управляющего графа; P_{S_3} – вероятность того, что нарушение привело к нетождественной модификации вычислительных операторов. Вероятности P_{S_1} и P_{S_2} можно оценить как:

$$P_{S_1} = \frac{\#X_i^- - \#X_i^- P_{\text{обни}}}{\#X_i} k_{\text{покр}} (1 - P_{\text{обни}}), \quad (6)$$

$$P_{S_2} = \frac{\#X_i^- - \#X_i^- P_{\text{обни}}}{\#X_i} (1 - k_{\text{покр}}), \quad (7)$$

где $\#$ – мощность множества.

Вероятность P_{S_3} – есть условная вероятность $P_{\text{мод}}$ модификации вычислительных операторов. Обозначим долю вредоносных входных данных как $k_{\text{вреди}} = \#X_i^- / \#X_i$. Тогда с учетом формул (6) и (7) получим:

$$P_{\text{проти}} = 1 - P_{\text{мод}} k_{\text{вреди}} (1 - P_{\text{обни}}) (1 - k_{\text{покр}} P_{\text{обни}}). \quad (8)$$

Поскольку эталонная семантическая модель программы формируется на основе ее управляющего графа, а ИТВ обнаруживается тогда и только тогда, когда оно приводит нарушению состояния или перехода, то вероятность ошибок I-го рода (ложных обнаружений) стремится к 0, что в целом характерно для

инвариантных методов [30]. Вероятность же ошибок II-го рода (ложных пропусков) определяется вероятностью пропуска ИТВ $P_{\text{нар}i}$.

В работе [35] под модификацией вычислений понимается злонамеренное нетождественное изменение арифметических операторов программы $Op\{+, -, *, /\}$ ($P_{Op_l|Op_j} = 0,25$). Если принять, что появление данных арифметических операторов в управляющем графе равновероятно ($P_{Op_j} = 0,25$), то вероятность возникновения модификации $P_{\text{мод}i}$ составит:

$$P_{\text{мод}i} = \sum_{j=1}^4 P_{Op_j} \left(\sum_{\substack{l=1 \\ j \neq l}}^4 P_{Op_l|Op_j} \right) = \sum_{j=1}^4 0,25 \left(\sum_{\substack{l=1 \\ j \neq l}}^4 0,25 \right) = 0,75.$$

Вероятность обнаружения нарушения $P_{\text{обн}i}$ зависит коэффициента $k_{\text{изв}i}$, определяющего то, какую часть составляют известные ИТВ на ОП КИИ. Допустим, известные ИТВ обнаруживаются всегда, а ранее неизвестные – с вероятностью, зависящей типа модификации. Арифметические операторы «+» и «-» неразличимы с точки зрения соотношений размерностей их операндов ($[L] = [P]$), а значит модификацию одного из них на другой невозможно обнаружить. Обозначим вероятность того, что модификация будет различима с точки зрения размерностей, как $p = 1 - (P_{+-} + P_{-+}) = 0,875$. Тогда вероятность обнаружения (а значит и восстановления) нарушений можно определить как:

$$P_{\text{обн}i} = P_{\text{восст}i} = k_{\text{изв}i} \cdot 1 + (1 - k_{\text{изв}i}) p. \quad (9)$$

Допускается, что время непосредственного вычисления программного цикла $T_{\text{выч}i}$ определяется временем выполнения всех вычислительных инструкций программы с учетом количества их параметров. Если также принять допущение, что параметры k_i , m_i и n_i одинаковы для всех программ i -го уровня ОП КИИ, то время вычислений можно оценить как $T_{\text{выч}i} = k_i m_i n_i t_0$.

Время, затрачиваемое дополнительно на контроль и обнаружение нарушений в процессе функционирования программы $T_{\text{обн}i}$, зависит от времени формирования и проверки матриц

размерностей в линейных блоках и пропорционально коэффициенту покрытия кибериммунитета $k_{\text{покр}i}$. Для формирования матриц размерностей (функция μ) потребуется для каждого из k_i блоков вычислить $m_i(m_i - 1)/2$ соотношений размерностей (для учета попарных связей) и выполнить $(2n_i - 1)$ элементарных операций взятия размерностей (обход дерева разбора арифметического выражения слева направо), после чего для обнаружения нарушения потребуется выполнить $k_i m_i n_i$ элементарных операций сравнения элементов текущей и эталонной матриц (функция η обнаружителя). Нарушение присутствует если один или более элементов матриц не совпадают. Тогда время $T_{\text{обн}i}$ можно определить как:

$$T_{\text{обн}i} = k_{\text{покр}i} \left(k_i \frac{m_i(m_i - 1)}{2} (2n_i - 1) + k_i m_i n_i \right) t_0. \quad (10)$$

Восстановление штатного функционирования осуществляется путем перезапуска или возврата к предыдущей контрольной точке. В обоих случаях необходимо выполнить переразметку программы в памяти путем выполнения элементарных операций чтения и записи всего образа программы (функция ξ восстановителя). Тогда время восстановления можно оценить как $T_{\text{восст}i} = 2k_i m_i n_i t_0$.

Поскольку ОП КИИ обладает вертикальными межуровневыми связями, то вероятность достижения цели функционирования ОП КИИ на i -ом уровне $P_{\text{цели}i}$ зависит от аналогичного показателя на $(i - 1)$ -ом уровне и с учетом формулы (4) определяется как:

$$P_{\text{цели}i} = P_{\text{цели}i-1} (P_{\text{проти}} + (1 - P_{\text{проти}}) P_{\text{восст}i}) P_{\text{раб}i}. \quad (11)$$

7. Теоретическое исследование модели. Исследуем системное свойство устойчивости ОП КИИ в условиях ИТВ на разработанной модели. Для этого сформулируем и докажем утверждение.

Утверждение 1. Увеличение значения коэффициента покрытия кибериммунитета $k_{\text{покр}i}$ повышает устойчивость ОП КИИ и время выполнения программ на i -ом уровне в условиях ИТВ.

Доказательство. Основные показатели устойчивости функционирования ОП КИИ определяются формулами (1)-(5), (8)-(11) и представляют собой непрерывные величины.

Вероятность противодействия ИТВ $P_{проти}$, в соответствии с формулой (8), монотонно возрастает при увеличении $k_{покрі}$.

Среднее время между пропусками ИТВ $T_{ср.атакі}$ определяется формулой (1) и зависит от времени вычислений $T_{вычі}$, времени обнаружения нарушений $T_{обні}$ и вероятности $P_{проти}$. При этом $T_{вычі}$ и $T_{воссті}$ не зависят от $k_{покрі}$, а $T_{обні}$, в соответствии с формулой (10), монотонно возрастает при увеличении $k_{покрі}$. Значит $T_{ср.атакі}$ также монотонно возрастает при увеличении $k_{покрі}$.

Вероятность восстановления $P_{воссті}$ (9) и среднее время восстановления $T_{ср.воссті}$ (2) не зависят от $k_{покрі}$.

Вероятность нахождения ОП КИИ в работоспособном состоянии $P_{рабі}$ определяется формулой (3), зависит от $T_{ср.атакі}$ и $T_{ср.воссті}$, и с учетом вышесказанного монотонно возрастает при увеличении $k_{покрі}$.

Вероятность достижения цели $P_{целиі}$ определяется формулой (11) и зависит от $P_{целиі-1}$, $P_{проти}$, $P_{воссті}$ и $P_{рабі}$. Пусть $P_{целиі-1} = 1$. Поскольку $P_{проти}$ и $P_{рабі}$ монотонно возрастают при увеличении $k_{покрі}$, а $P_{воссті} = const$ и $P_{воссті} < 1$, то $P_{целиі}$ монотонно возрастает при увеличении $k_{покрі}$.

Время выполнения программного цикла $T_{выпі}$ определяется формулой (5) и зависит от $T_{вычі}$, $T_{обні}$ и $T_{воссті}$. Поскольку $T_{обні}$ монотонно возрастает, а $T_{вычі}$ и $T_{воссті}$ не изменяются при увеличении $k_{покрі}$, то $T_{выпі}$ также монотонно возрастает при увеличении $k_{покрі}$.

Таким образом, все рассмотренные показатели устойчивости не уменьшаются, а основной показатель $P_{целиі}$ — увеличивается при увеличении $k_{покрі}$. Ч.т.д.

По результатам доказательства Утверждения 1 можно сделать следующий вывод. Увеличение $k_{\text{покр}i}$ повышает устойчивость ОП КИИ, но приводит к увеличению времени выполнения программ. В целом учет свойства кибериммунитета положительно влияет на устойчивость функционирования ОП КИИ в условиях ИТВ, а значит Гипотезу исследования можно считать *подтвержденной* теоретически.

8. Экспериментальное исследование модели. Целью эксперимента является исследование изменения показателей устойчивости ОП КИИ при варьировании $k_{\text{покр}i}$, определение существования минимального и максимального допустимых значений $k_{\text{покр}i}$ при наличии требований к показателям и проверка Гипотезы исследования. Зададим параметры моделирования: $P_{\text{цели}-1} = 1$, $P_{\text{мод}i} = 0,75$, $k_{\text{изв}i} = 0,4$, $k_i = 1000$, $m_i = 5$, $n_i = 2$.

На рисунке 5 представлены графики зависимостей показателей устойчивости функционирования ОП КИИ на i -ом уровне в условиях ИТВ от коэффициента покрытия кибериммунитета $k_{\text{покр}i}$. Как видно, при увеличении $k_{\text{покр}i}$ значения основных показателей устойчивости также возрастают. При этом следует отметить, что такие показатели как вероятность восстановления (рисунок 5(в)) и среднее время восстановления (рисунок 5(г)) не зависят от $k_{\text{покр}i}$ напрямую, однако при наличии кибериммунной защиты ($k_{\text{покр}i} > 0$) вероятность $P_{\text{восст}i}$ оказывается выше, а время $T_{\text{ср.восст}i}$ – ниже, чем при ее отсутствии ($k_{\text{покр}i} = 0$, $p = 0$).

Таким образом, увеличение $k_{\text{покр}i}$ положительно влияет на способность ОП КИИ противодействовать ИТВ, не допуская нарушений, и восстанавливать штатное функционирование при их возникновении, а учет свойства кибериммунитета в целом повышает устойчивость функционирования ОП КИИ, что согласуется с теоретическими выводами.

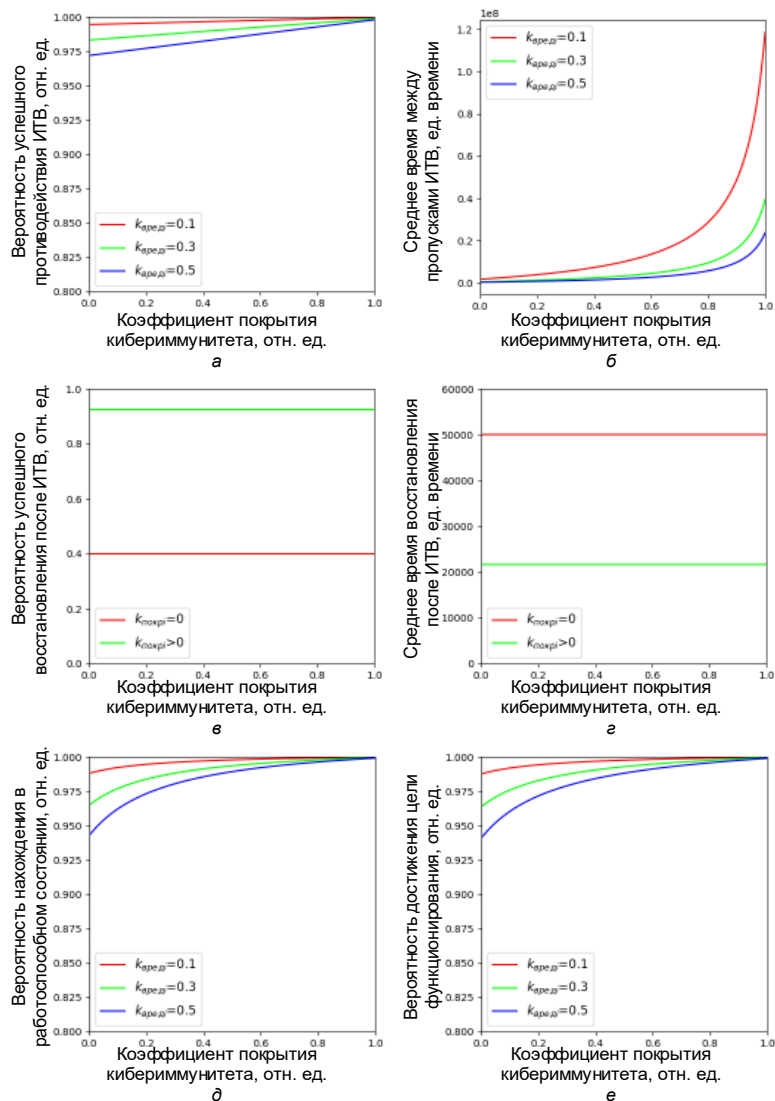


Рис. 5. Зависимости от коэффициента покрытия кибериммунитета $k_{\text{покр}}$:
 а – вероятности успешного противодействия ИТВ $P_{\text{проти}}$; б – ср. времени
 между пропусками ИТВ $T_{\text{ср.атаки}}$; в – вероятности успешного восстановления
 $P_{\text{восст}}$; г – ср. времени восстановления $T_{\text{ср.восст}}$; д – вероятности нахождения
 в работоспособном состоянии $P_{\text{раб}}$; е – вероятности достижения цели $P_{\text{цели}}$

Пусть теперь требуется, чтобы $P_{цели} \geq 0,96$ и $T_{вып} \leq 46000$ при $N = 4$ (количество уровней ОП КИИ). В таблице 1 представлены результаты исследования показателей устойчивости ОП КИИ для различных значений коэффициента $k_{врел}$. Отметим, что $P_{цели}$ характеризует вероятность достижения цели ОП КИИ на i -ом уровне, а $P_{цели}$ – общую вероятность достижения цели ОП КИИ. При этом $P_{цели} \propto P_{цели}^N$, поскольку нарушение на нижележащем уровне влияет на все вышележащие, а значит общая вероятность достижения цели ОП КИИ зависит от вероятности достижения ее на всех N уровнях.

Таблица 1. Результаты исследования показателей устойчивости функционирования ОП КИИ в условиях ИТВ

$k_{покр}$	$P_{прот}$	$T_{с.атак}$	$P_{восст}$	$T_{ср.восст}$	$P_{раб}$	$P_{цели}$	$T_{вып}$	$P_{цели}$
$k_{врел} = 0,1$								
0	0,9550	222222	0,4000	50000	0,8163	0,7943	30000	0,3980
0,0457	0,9946	2195568	0,9250	21622	0,9902	0,9898	31828	0,9600
0,2000	0,9954	3926380	0,9250	21622	0,9945	0,9942	38000	0,9769
0,4000	0,9965	7336861	0,9250	21622	0,9971	0,9968	46000	0,9872
0,5000	0,9970	9922480	0,9250	21622	0,9978	0,9976	50000	0,9904
0,8000	0,9985	28717948	0,9250	21622	0,9992	0,9991	62000	0,9965
$k_{врел} = 0,3$								
0	0,8650	74074	0,4000	50000	0,5970	0,5486	30000	0,0906
0,2000	0,9862	1308793	0,9250	21622	0,9837	0,9827	38000	0,9327
0,3787	0,9890	2293745	0,9250	21622	0,9907	0,9898	45148	0,9600
0,4000	0,9894	2445620	0,9250	21622	0,9912	0,9904	46000	0,9623
0,5000	0,9909	3307493	0,9250	21622	0,9935	0,9928	50000	0,9716
0,8000	0,9956	9572649	0,9250	21622	0,9977	0,9974	62000	0,9897
$k_{врел} = 0,5$								
0	0,7750	44444	0,4000	50000	0,4706	0,4071	30000	0,0275
0,2000	0,9771	785276	0,9250	21622	0,9732	0,9715	38000	0,8909
0,4000	0,9823	1467372	0,9250	21622	0,9855	0,9842	46000	0,9382
0,5000	0,9849	1984496	0,9250	21622	0,9892	0,9881	50000	0,9532
0,5541	0,9863	2346069	0,9250	21622	0,9909	0,9898	52164	0,9600
0,8000	0,9927	5743590	0,9250	21622	0,9962	0,9957	62000	0,9829

Видно, что при отсутствии кибериммунной защиты ($k_{покр} = 0$) ОП КИИ теряет устойчивость ($P_{цели} \rightarrow 0$). Увеличение же значения $k_{покр}$ повышает устойчивость ОП КИИ по всем обоснованным ранее показателям, однако это сопровождается увеличением времени

выполнения программ $T_{\text{вып}i}$. В случае $k_{\text{вред}i} = 0,1$ заданные требования обеспечиваются за счет варьирования значения коэффициента $k_{\text{покр}i}$ в диапазоне от 0,0457 до 0,4000, а в случае $k_{\text{вред}i} = 0,3$ – в диапазоне от 0,3787 до 0,4000. Иная ситуация наблюдается при $k_{\text{вред}i} = 0,5$. Здесь требование к $P_{\text{цели}}$ обеспечивается при $k_{\text{покр}i}$ не ниже 0,5541, однако для выполнения требования к $T_{\text{вып}i}$ допустимы значения $k_{\text{покр}i}$ не выше 0,4000, а значит в данном случае не существует значения $k_{\text{покр}i}$, при котором одновременно выполняются оба требования.

Полученная вероятность ошибок II-го рода (пропуск ИТВ) не превышает 0,03 (оценка на основе $\beta = 1 - P_{\text{проти}i}$), что согласуется с результатами исследований инвариантных методов обнаружения аномалий, приведенными в работе [30].

В дальнейшем целесообразно определить критерий существования решения, а также разработать методику синтеза оптимального значения $k_{\text{покр}i}$ с учетом требований к $P_{\text{цели}}$ и $T_{\text{вып}i}$.

Таким образом, результаты экспериментальных исследований устойчивости функционирования ОП КИИ в условиях ИТВ согласуются с теоретическими выводами, а значит, Гипотезу исследования можно считать *подтвержденной*.

9. Возможности практического применения результатов исследования. На рисунке 6 представлена возможная архитектура программного комплекса обеспечения устойчивости функционирования ОП КИИ.

Основными элементами программного комплекса являются:

- транслятор с SDK для внедрения кибериммунитета;
- подсистема управления устойчивостью, включающая модуль оценивания текущей устойчивости ОП КИИ и модуль управления параметрами кибериммунитета;
- подсистема кибериммунитета, включающая базу данных (БД) кибериммунитета, обнаружитель нарушений и восстановитель штатного функционирования.

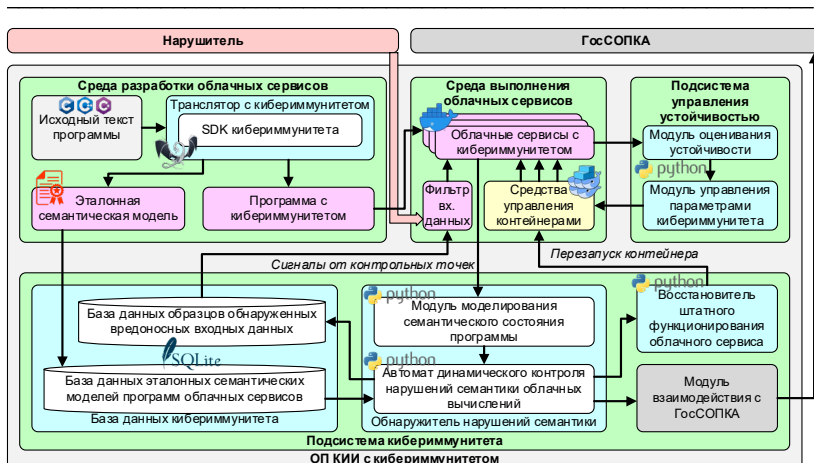


Рис. 6. Возможная архитектура программного комплекса обеспечения устойчивости функционирования ОП КИИ

Исходная программа облачного сервиса с помощью дополнений компилятора clang преобразуется в программу с кибериммунитетом путем внедрения элементов структурно-функциональной избыточности (меток начала и конца линейных блоков), при этом формируется ее эталонная модель в терминах формальных семантик (в частности, размерностей). Полученная программа с кибериммунитетом запускается в Docker-контейнере и циклически обрабатывает входные данные, находясь под воздействием ИТВ. В процессе функционирования программы сигналы о прохождении контрольных точек от внедренных меток передаются в обнаружитель нарушений, где текущее семантическое состояние программы в линейном блоке моделируется и сравнивается с эталонным (поэлементным сравнением матриц размерностей). При обнаружении нарушения состояния в результате атаки внедрения кода или нарушения перехода в результате атаки перехвата потока управления образец вредоносных входных данных заносится в БД SQLite и более не допускается. После этого восстановитель формирует команду перезапуска Docker-контейнера или возврата к предыдущему состоянию, если осуществляется их сохранение, и передает ее оркестратору. В процессе функционирования устойчивость ОП КИИ оценивается по показателям вероятности достижения цели и времени выполнения программного цикла и, при необходимости, осуществляется управление множеством контролируемых линейных блоков на основе коэффициента покрытия кибериммунитета.

В ходе накопления информации об отраженных ИТВ в БД SQLite могут сохраняться схожие образцы вредоносных входных данных одного формата. Поскольку данный формат неизвестен заранее, целесообразно осуществлять периодическую оптимизацию БД путем замены схожих образцов шаблонами. Имея несколько схожих образцов и применяя методы статистической индукции, можно своевременно нейтрализовать весь подкласс ранее неизвестных ИТВ, использующих вредоносные входные данные такого формата.

Таким образом, *практическая значимость* результатов исследования заключается в доведении их до технических рекомендаций по архитектуре программного комплекса. Данные технические рекомендации включают в себя разработку программных модулей, в совокупности реализующих новое эмерджентное свойство кибериммунитета, и могут быть применены для обеспечения устойчивости функционирования ОП КИИ, в частности «ГосТех».

10. Заключение. В работе предложена модель ОП КИИ с кибериммунитетом, позволяющая при принятых допущениях получать оценки введенных и обоснованных показателей устойчивости. *Научная новизна* модели заключается в том, что в нее впервые внедрены такие элементы, как обнаружитель нарушений семантики вычислений, восстановитель штатного функционирования и кибериммунная память. Данные элементы в совокупности реализуют новое эмерджентное свойство кибериммунитета, определяющее способность ОП КИИ противодействовать известным и новым, ранее неизвестным ИТВ, не допуская нарушений, и оперативно восстанавливать штатное функционирование при их возникновении.

Проведены теоретическое и экспериментальное исследования модели. Доказано утверждение о том, что увеличение значения коэффициента покрытия кибериммунитета повышает устойчивость функционирования ОП КИИ в условиях ИТВ по всем введенным показателям, но также приводит к увеличению времени выполнения программ. Для заданных условий экспериментально определены границы допустимых значений коэффициента покрытия кибериммунитета, в пределах которых одновременно обеспечиваются требования к вероятности достижения цели функционирования и времени выполнения программ ОП КИИ. По результатам исследований *подтверждена* выдвинутая Гипотеза исследования о том, что учет свойства кибериммунитета положительно влияет на устойчивость функционирования ОП КИИ в условиях ИТВ.

Практическая значимость результатов исследования заключается в доведении их до технических рекомендаций по

архитектуре программного комплекса, которые могут быть использованы при разработке средств защиты ОП КИИ, в частности, облачной платформы «ГосТех», в условиях ИТВ.

В *направления дальнейших исследований* входит: разработка методики синтеза оптимального значения коэффициента покрытия кибериммунитета с учетом требований к показателям устойчивости и времени выполнения программ, а также критерия его существования; исследование возможностей применения методов статистической индукции для превентивного реагирования на ранее неизвестные ИТВ на основе информации об известных вредоносных входных данных.

Литература

1. Официальный сайт компании CheckPoint. The State of Cyber Security 2025. URL: <https://www.checkpoint.com/security-report/> (дата обращения: 09.06.2025).
2. Официальный сайт компании «Солар». Тренды кибератак на промышленность и телеком в 2025 году. URL: <https://rt-solar.ru/analytics/reports/5522/> (дата обращения: 09.06.2025).
3. Официальный сайт «РИА Новости». Гендиректор «Солара» рассказал о росте числа кибератак на Россию. URL: <https://ria.ru/20250606/gk-2021325615.html> (дата обращения: 09.06.2025).
4. Зегжда Д.П., Александрова Е.Б., Калинин М.О., и др. Кибербезопасность цифровой индустрии. Теория и практика функциональной устойчивости к кибератакам // Москва: Научно-техническое издательство «Горячая линия-Телеком». 2021. 560 с.
5. Павленко Е.Ю., Штыркина А.А., Зегжда Д.П. Оценка устойчивости киберфизических систем на основе спектральной теории графов // Проблемы информационной безопасности. Компьютерные системы. 2019. № 1. С. 60–68.
6. Саенко И.Б., Котенко И.В., Лаута О.С., Скоробогатов С.Ю. Методика оценки устойчивости программно-конфигурируемых сетей в условиях компьютерных атак // I-methods. 2023. Т. 15. № 1.
7. Саенко И.Б., Котенко И.В., Лаута О.С., Скоробогатов С.Ю. Модели компьютерных атак на программно-конфигурируемые сети // Научные технологии в космических исследованиях Земли. 2023. Т. 15. № 1. С. 37–47. DOI: 10.36724/2409-5419-2023-15-1-37-47.
8. Бирюков Д.Н., Ломачко А.Г., Ростовцев Ю.Г. Облик антиципирующих систем предотвращения рисков реализации киберугроз // Труды СПИИРАН. 2015. № 2(39). С. 5–25. DOI: 10.15622/sp.39.1.
9. Андрушкевич Д.В., Бирюков Д.Н., Тимашов П.В. Порождение сценариев предотвращения компьютерных атак на основе логико-онтологического подхода // Труды Военно-космической академии имени А.Ф. Можайского. 2021. № 677. С. 118–134.
10. Кубрин Г.С., Зегжда Д.П. Выявление дефектов в многокомпонентном программном обеспечении с применением набора универсальных графовых представлений кода // Проблемы информационной безопасности. Компьютерные системы. 2024. № S2(60). С. 65–75. DOI: 10.48612/jisp/nb67-m5g8-mpae.

11. Chevtchenko S.F., et al. Anomaly Detection in Industrial Machinery Using IoT Devices and Machine Learning: A Systematic Mapping // IEEE Access. 2023. vol. 11. pp. 128288–128305. DOI: 10.1109/ACCESS.2023.3333242.
12. Nand K., Zhang Z., Hu J. A Comprehensive Survey on the Usage of Machine Learning to Detect False Data Injection Attacks in Smart Grids // IEEE Open Journal of the Computer Society. 2025. vol. 6. pp. 1121–1132. DOI: 10.1109/OJCS.2025.3585248.
13. Hao W., Yang T., Yang Q. Hybrid Statistical-Machine Learning for Real-Time Anomaly Detection in Industrial Cyber-Physical Systems // IEEE Transactions on Automation Science and Engineering. 2023. vol. 20. no. 1. pp. 32–46. DOI: 10.1109/TASE.2021.3073396.
14. Ozdogan E. A Comprehensive Analysis of the Machine Learning Algorithms in IoT IDS Systems // IEEE Access. 2024. vol. 12. pp. 46785–46811. DOI: 10.1109/ACCESS.2024.3382539.
15. Новикова Е.С., Котенко И.В., Мелешко А.В., Израилев К.Е. Обнаружение вторжений на основе федеративного обучения: архитектура системы и эксперименты // Вопросы кибербезопасности. 2023. № 6(58). С. 50–66. DOI: 10.21681/2311-3456-2023-6-50-66.
16. Aljuaid W.H., Alshamrani S.S. A deep learning approach for intrusion detection systems in cloud computing environments // Applied sciences. 2024. vol. 14. no. 13. DOI: 10.3390/app14135381.
17. Alrayes F.S., Zakariah M., Amin S.U., Iqbal Khan Z., Helal M. Intrusion Detection in IoT Systems Using Denoising Autoencoder // IEEE Access. 2024. vol. 12. pp. 122401–122425. DOI: 10.1109/ACCESS.2024.3451726.
18. Liu X., Xie L., Wang Y., Zou J., Xiong J., Ying Z. Privacy and Security Issues in Deep Learning: A Survey // IEEE Access. 2021. vol. 9. pp. 4566–4593. DOI: 10.1109/ACCESS.2020.3045078.
19. Бурлаков М.Е., Ивкин А.Н. Система обнаружения вторжения на основе искусственной иммунной системы // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2019. № 29. С. 209–224.
20. Шамсутдинов Р.Р., Васильев В.И., Вульфин А.М. Интеллектуальная система мониторинга информационной безопасности промышленного интернета вещей с использованием механизмов искусственных иммунных систем // Системная инженерия и информационные технологии. 2024. Т. 6. № 4(19). С. 14–31. DOI: 10.54708/2658-5014-SIT-2024-но4-р14.
21. Браницкий А.А., Котенко И.В. Обнаружение сетевых атак на основе комплексирования нейронных, иммунных и нейронечетких классификаторов // Информационно-управляющие системы. 2015. № 4(77). С. 69–77. DOI: 10.15217/issn1684-8853.2015.4.69.
22. Dutt I., Borah S., Maitra I.K. Immune System Based Intrusion Detection System (IS-IDS): A Proposed Model // IEEE Access. 2020. vol. 8. pp. 34929–34941. DOI: 10.1109/ACCESS.2020.2973608.
23. Aldhaheri S., Alghazzawi D., Cheng L., Alzahrani B., Al-Barakati A. DeepDCA: Novel Network-Based Detection of IoT Attacks Using Artificial Immune System // Appl. Sci. 2020. vol. 10(6). DOI: 10.3390/app10061909.
24. Gijzen B., Montalto R., Panneman J., Falconieri F., Wiper P., Zuraniewski P. Self-Healing for Cyber-Security // Sixth International Conference on Fog and Mobile Edge Computing (FMEC). 2021. pp. 1–7. DOI: 10.1109/FMEC54266.2021.9732575.
25. Pinto C., Pinto R., Gonçalves G. Towards Bio-Inspired Anomaly Detection Using the Cursory Dendritic Cell Algorithm // Algorithms. 2022. vol. 15(1). DOI: 10.3390/al15010001.

26. Bereta M. Negative selection algorithm for unsupervised anomaly detection // *Applied sciences*. 2024. vol. 14. no. 23. DOI: 10.3390/app142311040.
27. Jerbi M., Dagdia Z.C., Bechikh S., Said L.B. Immune-based system to enhance malware detection // *IEEE congress on evolutionary computation (CEC)*. 2023. pp. 1–8. DOI: 10.1109/CEC53210.2023.10254159.
28. Воеводин В.А. О постановке задачи оценивания устойчивости функционирования объектов критической информационной инфраструктуры // *Вопросы кибербезопасности*. 2025. № 1(65). С. 41–49. DOI: 10.21681/2311-3456-2025-1-41-49.
29. Балябин А.А., Петренко С.А. Модель самовосстановление киберфизических систем КИИ РФ в условиях кибератак на основе кибериммунитета // *Сборник трудов IX Международной научно-технической конференции (CDE'25)*. 2025. С. 76–91.
30. Петренко С.А. Кибериммунология: научная монография // Санкт-Петербург: Издательский дом «Афина». 2021. 240 с.
31. Петренко С.А. Киберустойчивость индустрии 4.0: научная монография // Санкт-Петербург: Издательский дом «Афина». 2020. 256 с.
32. Balyabin A.A. Threats to the Resilience of Cloud Platforms // *XXVII International Conference on Soft Computing and Measurements (SCM)*. 2024. pp. 246–249. DOI: 10.1109/SCM62608.2024.10554080.
33. Balyabin A.A. Ensuring the Resilience of Cloud Platforms Based on Cyber Immunity // *XXVII International Conference on Soft Computing and Measurements (SCM)*. 2024. pp. 233–237. DOI: 10.1109/SCM62608.2024.10554277.
34. Балябин А.А. Модель облачной платформы КИИ РФ с кибериммунитетом в условиях информационно-технических воздействий // *Защита информации. Инсайд*. 2024. № 5(119). С. 35–44.
35. Харжевская А.В., Ломако А.Г., Петренко С.А. Представление программ инвариантами подобия для контроля искажения вычислений // *Вопросы кибербезопасности*. 2017. № 2(20). С. 9–20. DOI: 10.21581/2311-3456-2017-2-9-20.

Балябин Артём Алексеевич — младший научный сотрудник, научный центр информационных технологий и искусственного интеллекта, Научно-технологический университет "Сириус". Область научных интересов: исследование безопасности программного обеспечения, формальная верификация, обратный инжиниринг программных комплексов, организация самовосстанавливающихся вычислений, кибериммунология, искусственный интеллект, блокчейн, квантовые вычисления. Число научных публикаций — 112. balyabin.aa@talantiuspeh.ru; проспект Олимпийский, 1, 354340, Федеральная территория "Сириус", Россия; р.т.: +7(911)260-0620.

Петренко Сергей Анатольевич — д-р техн. наук, профессор, руководитель группы, научный центр информационных технологий и искусственного интеллекта, Научно-технологический университет "Сириус". Область научных интересов: квантовая информатика и информационная безопасность, информационные технологии и искусственный интеллект. Число научных публикаций — 560. petrenko.sa@talantiuspeh.ru; проспект Олимпийский, 1, 354340, Федеральная территория "Сириус", Россия; р.т.: +7(903)742-8543.

Поддержка исследований. Исследование выполнено за счет гранта Российского научного фонда (№ 25-11-20037, <https://www.rscf.ru/project/25-11-20037/>).

A. Balyabin, S. Petrenko
**MODEL OF A CRITICAL INFORMATION INFRASTRUCTURE
CLOUD PLATFORM WITH CYBER IMMUNITY**

Balyabin A., Petrenko S. Model of a Critical Information Infrastructure Cloud Platform with Cyber Immunity.

Abstract. The research is devoted to solving the problem of synthesizing a model of a critical information infrastructure cloud platform with cyber immunity. The relevance of the research is due to the need to resolve a problematic situation characterized by contradictions in science and practice. The contradiction in practice is observed between increased requirements for the resilience of critical information infrastructure cloud platforms and the growth of threats associated with the exploitation of previously unknown vulnerabilities and the overcoming of protective measures. The contradiction in science is that it is impossible to ensure the required resilience of such platforms using existing models and methods. Thus, existing approaches do not fully account for the specific features of critical information infrastructure cloud platforms, such as hierarchical architecture, the presence of undetected vulnerabilities, operation under targeted cyberattacks, increased requirements for resilience, and the need for rapid restoration of normal operation. This paper aims to synthesize a new model of a critical information infrastructure cloud platform with cyber immunity. A hypothesis has been formulated that endowing cloud platforms with the property of cyber immunity has a positive effect on their resilience when subjected to cyberattacks. Research methods include methods of system analysis, probability theory, theory of formal semantics, theory of similarity and dimensional analysis, as well as cyber immunology methods. The concept of cyber immunity has been substantiated, which involves providing cloud platforms with the ability to counteract known and previously unknown cyberattacks, quickly restore normal operation, and memorize malicious input data, thereby preventing their processing in the future. The indicators of the resilience of critical information infrastructure cloud platforms have also been substantiated. A new model of a critical information infrastructure cloud platform with cyber immunity has been developed. The scientific novelty of the proposed model lies in the introduction, for the first time, of components such as a semantic violation detector, a normal operation restorer, and cyber immune memory. These components collectively implement a new emergent property of cyber immunity. Theoretical and experimental studies of the model have been conducted, confirming the proposed hypothesis. The practical significance of the research results lies in providing technical recommendations on the architecture of the software complex, which can be applied in the development of means for protecting critical information infrastructure cloud platforms, in particular, the GosTech cloud platform, against cyberattacks.

Keywords: cloud computing, computation semantics, critical information infrastructure, cyber resilience, cyberattacks, cyber immunity.

References

1. Официальный сайт компании CheckPoint. The State of Cyber Security 2025. Available at: <https://www.checkpoint.com/security-report/> (accessed 09.06.2025). (In Russ.).
2. Официальный сайт компании «Солар». Тренды кибератак на промышленность и телеком в 2025 году. Available at: <https://rt-solar.ru/analytics/reports/5522/> (accessed 09.06.2025). (In Russ.).

3. Официальный сайт «РИА Новости». Гендиректор «Солара» рассказал о росте числа кибератак на Россию. Available at: <https://ria.ru/20250606/gk-2021325615.html> (accessed 09.06.2025). (In Russ.).
4. Zegzhda D.P., Aleksandrova E.B., Kalinin M.O., et al. Kiberbezopasnost' tsifrovoy industrii. Teoriya i praktika funktsional'noy ustoychivosti k kiberatakam [Cybersecurity of the Digital Industry: Theory and Practice of Functional Resilience to Cyberattacks]. Moscow: Nauchno-tekhnicheskoe izdatel'stvo «Goryachaya liniya-Telekom». 2021. 560 p. (In Russ.).
5. Pavlenko E.Yu., Shtyrkina A.A., Zegzhda D.P. [Estimating the Cyber-Physical System Sustainability Based on Spectral Graph Theory]. Problemy informatsionnoy bezopasnosti. Komp'yuternye systemy – Problems of information security. Computer systems. 2019. no. 1. pp. 60–68. (In Russ.).
6. Saenko I.B., Kotenko I.V., Lauta O.S., Skorobogatov S.Yu. [Sustainability Assessment Methodology Software-Configurable Networks in the Conditions of Computer Attacks]. I-methods. 2023. vol. 15. no. 1. (In Russ.).
7. Saenko I.B., Kotenko I.V., Lauta O.S., Skorobogatov S.Yu. [Computer Attack Models on Software-Configurable Networks]. Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli – High technologies in earth space research. 2023. vol. 15. no. 1. pp. 37–47. DOI: 10.36724/2409-5419-2023-15-1-37-47. (In Russ.).
8. Biryukov D., Lomako A., Rostovtsev Y. The Appearance of Anticipating Cyber Threats Risk Prevention Systems. SPIIRAS Proceedings. Труды СПИИРАН. 2015. no. 2(39). pp. 5–25. DOI: 10.15622/sp.39.1. (In Russ.).
9. Andrushkevich D.V., Biryukov D.N., Timashov P.V. [Synthesis of Computer Attack Prevention Scenarios Based on a Logical-Ontological Approach]. Trudy Voenno-kosmicheskoy akademii imeni A.F. Mozhayskogo – Proceedings of the A.F. Mozhaysky Military Space Academy. 2021. no. 677. pp. 118–134. (In Russ.).
10. Kubrin G.S., Zegzhda D.P. [Vulnerability Detection in Multicomponent Software Using a Set of Generalized Code Graph Representations]. Problemy informatsionnoy bezopasnosti. Komp'yuternye systemy – Problems of information security. Computer systems. 2024. no. S2(60). pp. 65–75. DOI: 10.48612/jisp/nb67-m5g8-mpae. (In Russ.).
11. Chevchenko S.F., et al. Anomaly Detection in Industrial Machinery Using IoT Devices and Machine Learning: A Systematic Mapping. IEEE Access. 2023. vol. 11. pp. 128288–128305. DOI: 10.1109/ACCESS.2023.3333242.
12. Nand K., Zhang Z., Hu J. A Comprehensive Survey on the Usage of Machine Learning to Detect False Data Injection Attacks in Smart Grids. IEEE Open Journal of the Computer Society. 2025. vol. 6. pp. 1121–1132. DOI: 10.1109/OJCS.2025.3585248.
13. Hao W., Yang T., Yang Q. Hybrid Statistical-Machine Learning for Real-Time Anomaly Detection in Industrial Cyber-Physical Systems. IEEE Transactions on Automation Science and Engineering. 2023. vol. 20. no. 1. pp. 32–46. DOI: 10.1109/TASE.2021.3073396.
14. Ozdogan E. A Comprehensive Analysis of the Machine Learning Algorithms in IoT IDS Systems. IEEE Access. 2024. vol. 12. pp. 46785–46811. DOI: 10.1109/ACCESS.2024.3382539.
15. Novikova E.S., Kotenko I.V., Meleshko A.V., Izrailov K.E. [Federated Learning Based Intrusion Detection: System Architecture and Experiments]. Voprosy kiberbezopasnosti – Cybersecurity issues. 2023. no. 6(58). pp. 50–66. DOI: 10.21681/2311-3456-2023-6-50-66. (In Russ.).
16. Aljuaid W.H., Alshamrani S.S. A deep learning approach for intrusion detection systems in cloud computing environments. Applied sciences. 2024. vol. 14. no. 13. DOI: 10.3390/app14135381.

17. Alrayes F.S., Zakariah M., Amin S.U., Iqbal Khan Z., Helal M. Intrusion Detection in IoT Systems Using Denoising Autoencoder. IEEE Access. 2024. vol. 12. pp. 122401–122425. DOI: 10.1109/ACCESS.2024.3451726.
18. Liu X., Xie L., Wang Y., Zou J., Xiong J., Ying Z. Privacy and Security Issues in Deep Learning: A Survey. IEEE Access. 2021. vol. 9. pp. 4566–4593. DOI: 10.1109/ACCESS.2020.3045078.
19. Burlakov M.E., Ivkin A.N. [Intrusion Detection System Based on the Artificial Immune System]. Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniya – Bulletin of Perm National Research Polytechnic University. Electrical Engineering, Information Technologies, and Control Systems. 2019. no. 29. pp. 209–224. (In Russ.).
20. Shamsutdinov R.R., Vasil'ev V.I., Vul'fin A.M. [Intelligent System for Monitoring Information Security of the Industrial Internet of Things using Artificial Immune Systems Mechanisms]. Sistemnaya inzheneriya i informatsionnye tekhnologii – Systems engineering and information technologies. 2024. vol. 6. no. 4(19). pp. 14–31. DOI: 10.54708/2658-5014-SIIT-2024-no4-p14. (In Russ.).
21. Branitskiy A.A., Kotenko I.V. [Network Attack Detection Based on Combination of Neural, Immune and Neuro-Fuzzy Classifiers]. Informatsionno-upravlyayushchie sistemy – Information and control systems. 2015. no. 4(77). pp. 69–77. DOI: 10.15217/issn1684-8853.2015.4.69. (In Russ.).
22. Dutt I., Borah S., Maitra I.K. Immune System Based Intrusion Detection System (IS-IDS): A Proposed Model. IEEE Access. 2020. vol. 8. pp. 34929–34941. DOI: 10.1109/ACCESS.2020.2973608.
23. Aldhaheri S., Alghazzawi D., Cheng L., Alzahrani B., Al-Barakati A. DeepDCA: Novel Network-Based Detection of IoT Attacks Using Artificial Immune System. Appl. Sci. 2020. vol. 10(6). DOI: 10.3390/app10061909.
24. Gijsen B., Montalto R., Panneman J., Falconieri F., Wiper P., Zuraniewski P. Self-Healing for Cyber-Security. Sixth International Conference on Fog and Mobile Edge Computing (FMEC). 2021. pp. 1–7. DOI: 10.1109/FMEC54266.2021.9732575.
25. Pinto C., Pinto R., Gonçalves G. Towards Bio-Inspired Anomaly Detection Using the Cursory Dendritic Cell Algorithm. Algorithms. 2022. vol. 15(1). DOI: 10.3390/a15010001.
26. Bereta M. Negative selection algorithm for unsupervised anomaly detection. Applied sciences. 2024. vol. 14. no. 23. DOI: 10.3390/app142311040.
27. Jerbi M., Dagdia Z.C., Bechikh S., Said L.B. Immune-based system to enhance malware detection. IEEE congress on evolutionary computation (CEC). 2023. pp. 1–8. DOI: 10.1109/CEC53210.2023.10254159.
28. Voevodin V.A. [On the formulation of the task of assessing the stability of the functioning of critical information infrastructure facilities]. Voprosy kiberbezopasnosti – Cybersecurity issues. 2025. no. 1(65). pp. 41–49. DOI: 10.21681/2311-3456-2025-1-41-49. (In Russ.).
29. Balyabin A.A., Petrenko S.A. [A self-healing model for cyber-physical systems of the Russian Federation's critical information infrastructure under cyberattacks based on cyber-immunity]. Sb. tr. IX Mezhdunar. nauchn.-tekhn. konf. (CDE'25) [The 2025 Symposium on Cybersecurity of the Digital Economy (CDE'25): Collected papers]. 2025. pp. 76–91. (In Russ.).
30. Petrenko S.A. Kiberimmunologiya: nauchnaya monografiya [Cyber Immunology]. Saint Petersburg: Izdatel'skiy dom «Afina». 2021. 239 p. (In Russ.).
31. Petrenko S.A. Kiberustoychivost' industrii 4.0 [Cyber Resilience Industry 4.0]. Saint Petersburg: Izdatel'skiy dom «Afina». 2020. 256 p. (In Russ.).

32. Balyabin A.A. Threats to the Resilience of Cloud Platforms. XXVII International Conference on Soft Computing and Measurements (SCM). 2024. pp. 246–249. DOI: 10.1109/SCM62608.2024.10554080.
33. Balyabin A.A. Ensuring the Resilience of Cloud Platforms Based on Cyber Immunity. XXVII International Conference on Soft Computing and Measurements (SCM). 2024. pp. 233–237. DOI: 10.1109/SCM62608.2024.10554277.
34. Balyabin A.A. [Model of the Cloud Platform of Critical IT Infrastructure of the Russian Federation Under the Conditions of Information Technology Impacts]. *Zašita informacii. Inside – Information protection. Inside*. 2024. no. 5(119). pp. 35–44. (In Russ.).
35. Kharzhevskaya A., Lomako A., Petrenko S. [Representing programs with similarity invariants for monitoring tampering with calculations]. *Voprosy kiberbezopasnosti – Cybersecurity issues*. 2017. no. 2(20). pp. 9–20. DOI: 10.21581/2311-3456-2017-2-9-20. (In Russ.).

Balyabin Artyom — Junior researcher, Scientific center for information technologies and artificial intelligence, Sirius University of Science and Technology. Research interests: software security research, formal verification, reverse engineering of software systems, organization of self-healing computations, cyber immunology, artificial intelligence, blockchain, quantum computing. The number of publications — 112. balyabin.aa@talantiuspeh.ru; 1, Olympiysky Av., 354340, Federal Territory "Sirius", Russia; office phone: +7(911)260-0620.

Petrenko Sergei — Ph.D., Dr.Sci., Professor, Leader of the group, Scientific center for information technologies and artificial intelligence, Sirius University of Science and Technology. Research interests: quantum informatics and information security, information technology and artificial intelligence. The number of publications — 560. petrenko.sa@talantiuspeh.ru; 1, Olympiysky Av., 354340, Federal Territory "Sirius", Russia; office phone: +7(903)742-8543.

Acknowledgements. This research is supported by the Russian Science Foundation (project № 25-11-20037, <https://www.rscf.ru/project/25-11-20037/>).

П.Д. БОРИСОВ, Д.В. ВАРЛАМОВ, Ю.В. КОСОЛАПОВ
**АЛГОРИТМ ВЫЧИСЛЕНИЯ ПОХОЖЕСТИ ГРАФОВ И ЕГО
ПРИМЕНЕНИЕ ДЛЯ СРАВНЕНИЯ БИНАРНЫХ
ИСПОЛНЯЕМЫХ ФАЙЛОВ**

Борисов П.Д., Варламов Д.В., Косолапов Ю.В. Алгоритм вычисления похожести графов и его применение для сравнения бинарных исполняемых файлов.

Аннотация. Рассматривается задача статического (без запуска) сравнения бинарных исполняемых файлов. Программа и любая ее процедура могут быть представлены в виде ориентированного графа. Для программы соответствующий граф представляет собой граф вызова функций (процедур), где узлами являются сами функции, а ребро из вершины a в b описывает вызов функции b из функции a . Для процедуры такой граф представляет собой граф потока управления, где вершинами являются базовые блоки, а ребро между узлами a и b означает возможное исполнение команд блока b после исполнения команд блока a . В работе предлагается алгоритм сравнения направленных графов, который далее применяется для сравнения программ. В основе алгоритма сравнения графов лежит применение функции похожести узлов. Для сравнения графов процедур в качестве такой функции похожести применяются нечеткая (fuzzy) хеш-функция и криптографическая хеш-функция. Далее этот способ сравнения графов процедур используется как функция похожести узлов при сравнении графов программ. На базе предложенного алгоритма разработан метод сравнения программ, проведено его исследование в рамках двух экспериментов. В первом эксперименте исследовано поведение метода при сравнении программ, полученных с применением разных опций оптимизации (O0, O1, O2, O3 и Os). Во втором эксперименте исследована возможность выявления эффективных и стойких обфусцирующих преобразований в рамках ранее разработанной модели. В первом эксперименте получены свидетельства в пользу верности гипотезы об уменьшении похожести файлов с ростом оптимизации от O1 до O3. Во втором эксперименте подтверждены некоторые полученные ранее результаты, касающиеся эффективности (неэффективности) и стойкости (нестойкости) обфусцирующих преобразований.

Ключевые слова: сравнение графов, похоть программ, эффективность и стойкость обфускации.

1. Введение. В области защиты информации *различители* играют важную роль. В общем виде различитель можно определить как функцию dis , принимающую в качестве аргумента два объекта, отличие в которых находится, и возвращающую значение из отрезка действительных чисел $[0, 1]$. Чем ближе значение, возвращаемое различителем, к 1,

тем больше отличие в сравниваемых объектах, и наоборот (нулевое значение соответствует полному совпадению сравниваемых объектов в выбранном пространстве признаков). Например, в криптографии наличие эффективного различителя, позволяющего отличить (с пренебрежимой вероятностью) вариант использования криптографической схемы, когда в ней задействуется «неслучайный» криптографический элемент/примитив (например, помехоустойчивый код), от варианта, когда применяется «случайный» аналог этого примитива, часто свидетельствует о слабости этой криптографической схемы [1]. В стеганографии стойкость стегосистемы (стойкость к обнаружению скрытого канала передачи информации) зависит от того, насколько вычислительно сложно отличить стегообъект (объект, содержащий скрытые данные) от чистого покрывающего объекта (объекта, не содержащего встроенных данных) [2]. Решение задачи распознавания помехоустойчивого кода, используемого в канале связи, также часто основывается на построении различителя [3]. В компьютерной безопасности системы защиты от спама, атак и от вредоносного кода обычно строятся на основе различителей. Именно, при защите от спама используются алгоритмы отличия входящей корреспонденции (электронных писем, мгновенных сообщений), относящейся к нежелательной (к спаму), от остальной [4]. При обнаружении атак различители позволяют отличить «нормальное» состояние (состояние сети или состояние кода, исполняемого в текущий момент на процессоре) от аномального [5–7]. А при проверке файлов на наличие вредоносного кода различители используются как дополнительный механизм выявления ранее неизвестного или замаскированного вредоносного кода, когда проверка по базе сигнатур не дает результата [8, 9]. В [10] предложен способ оценки эффективности и стойкости обфусцирующих преобразований программ, в основе которого лежит применение функции похожести sim , определённой на парах исполняемых файлов и возвращающей значение от 0 до 1. На основе этой функции естественным образом может быть построен различитель dis для произвольной пары программ P_1 и P_2 : $\text{dis}(P_1, P_2) = 1 - \text{sim}(P_1, P_2)$. Поэтому естественно считать, что модель из [11] основана на применении различителя, определённого на парах программ.

В настоящей работе предлагается способ построения функции sim , которая может служить, с одной стороны, как дополнительный показатель похожести в рамках функции похожести программ, предложенной в [11] на основе методов машинного обучения, с другой – может применяться самостоятельно при оценке структурной похожести файлов исполняемых программ.

Работа, кроме введения и заключения, содержит четыре раздела. Во втором разделе кратко приводится обзор подходов к сравнению файлов. Третий раздел посвящен построению алгоритма сравнения ориентированных графов. На основе этого алгоритма в четвертом разделе строится метод сравнения исполняемых файлов. Экспериментальному исследованию этого метода посвящен пятый раздел.

2. Обзор литературы. Как отмечено в [11], условно алгоритмы сравнения программ можно разделить на *универсальные* и *специализированные*. В первом случае алгоритмы сравнения не используют информацию о типе сравниваемых файлов: эти файлы рассматриваются как последовательности байтов. К таким относятся, например, алгоритмы нечетких хеш-функций *ssdeep* [12], *mrsh-v2* [13], *tlsh* и *sdhash* [14]. Независимость от типа сравниваемых файлов позволяет применять эти функции в различных сценариях обработки информации, в частности, в сценариях защиты от спама, при поиске вредоносного кода, похожего на уже известный код и т.п. В [15] показано, что при сравнении исполняемых программ указанные выше функции показывают низкую точность обнаружения похожих файлов. Например, изменение одной ассемблерной инструкции может привести к падению до нуля значения похожести. Специализированные алгоритмы сравнения, в свою очередь, учитывают структуру исполняемых файлов, поэтому применимы только к файлам заданного типа. Однако, с другой стороны, это позволяет сохранить высокую точность сравнения при незначительных изменениях файлов. К таким алгоритмам можно отнести *bindiff* [16], *bcc* [17], *machos* [18], *machoke* [19] и *tah* [20], которые основаны на сравнении графов процедур. В частности, алгоритмы *bindiff* и *bcc* находят похожие процедуры в сравниваемых бинарных файлах путем применения ряда эвристик (при сопоставлении графов процедур), позволяющих судить о семантической эквивалентности сравниваемых процедур. К таким эвристикам, в частности, относится сравнение хеш-значений сравниваемых процедур. Дополнительное использование в *bcc* графов зависимостей от данных и управления (PDG – program dependence graph) позволяет повысить долю верно найденных семантически эквивалентных процедур, по сравнению с *bindiff*. В алгоритмах *machos* и *machoke* для сравниваемых процедур во время обхода соответствующих графов строятся последовательности хеш-значений базовых блоков, после чего вычисляются финальные хеш-значения от этих последовательностей и выполняется их сравнение. Стоит отметить, что алгоритмы *machos* и *machoke* отличаются только способами получения графовых представлений процедур: в первом

случае используется дизассемблер IDA Pro, а во втором – Radare2. Алгоритм *tah* также строит хеш-значение процедуры, основываясь на ее графе, однако, строит это значение, основываясь не на содержимом базовых блоков, а на основе связей между блоками.

В [11] экспериментально показано, что при совместном использовании специализированных и универсальных функций похожести в моделях сравнения файлов на основе методов машинного обучения наибольший вклад в принятие решения дают специализированные функции. Таким образом, пополнение арсенала специализированных функций похожести может позволить, например, выполнять более точную оценку эффективности и стойкости обфусцирующих преобразований в рамках модели, предложенной в [10].

3. Алгоритм сравнения графов. Программы и процедуры (далее функции, из которых состоит программа, будут называться процедурами, чтобы исключить конфликт с термином «функция похожести») часто представляются направленными графами. Под направленным графом далее будем понимать пару $G = \langle V, E \rangle$ с множеством вершин V и множеством ребер E . Естественной представляется разработка алгоритма сравнения абстрактных графов $G^1 = \langle V^1, E^1 \rangle$ и $G^2 = \langle V^2, E^2 \rangle$ с последующим применением его для сравнения программ. Далее описывается алгоритм sim_G сравнения графов (алгоритм 1), идея которого предложена в [21].

Алгоритм 1. Алгоритм сравнения графов sim_G

Вход: $G^i = \langle V^i, E^i \rangle$, sim_v

Выход: $s \in [0, 1]$

- 1: Построить список L вида (1); $l = 0$
 - 2: **while** $L \neq \emptyset$ **do**
 - 3: Выбрать первый элемент $(v, \tilde{v}, c)$ из L .
 - 4: $\text{tag}_{G^1}(v) \leftarrow l$, $\text{tag}_{G^2}(\tilde{v}) \leftarrow l$.
 - 5: Удалить из L все тройки $(v', \tilde{v}', c')$, в которых $v' = v$ или $\tilde{v}' = \tilde{v}$.
 - 6: $l = l + 1$
 - 7: **end while**
 - 8: $i = \text{argmax}_{k \in \{1, 2\}} |V^k|$, $r = ||V^1| - |V^2||$
 - 9: Назначить узлам графа G^i в произвольном порядке метки с номерами от l до $l + r - 1$.
 - 10: Добавить в граф G^{3-i} фиктивные узлы без связей; присвоить этим узлам уникальные значения начальных меток, а новым меткам этих вершин присвоить значения от l до $l + r - 1$.
 - 11: По графу G^i построить матрицу смежности вершин $M^i = M(G^i)$, $i = 1, 2$, в которой строки и столбцы упорядочены по возрастанию значений новых меток от 0 до $l + r - 1$.
 - 12: Найти $\rho(G^1, G^2)$ в соответствии с (4).
- return** $\rho(G^1, G^2)$.
-

Алгоритм sim_G сравнения графов G^1 и G^2 основан на применении функции похоти $\text{sim}_v : V^1 \times V^2 \rightarrow [0, 1]$. Значение $\text{sim}_v(v, \tilde{v})$ можно рассматривать как похоть узла $v \in V^1$ на узел $\tilde{v} \in V^2$: чем ближе значение к 1, тем узлы более похоты, и наоборот. Предполагается, что узлы сравниваемых графов имеют две метки: начальную метку и новую метку. Начальная метка узла v совпадает с v , а новую метку графа G будем обозначать $\text{tag}_G(v)$. В алгоритме 1 узлы двух графов сопоставляются на основе функции sim_v . Сначала по графам G^1 и G^2 строится список троек

$$L = \{(v, \tilde{v}, c) : (v, \tilde{v}) \in V^1 \times V^2, c = \text{sim}_v(v, \tilde{v})\}, \quad (1)$$

упорядоченный по убыванию значения c . Таким образом, первый элемент этого списка будет содержать начальные метки v и $\tilde{v}$ двух вершин (одной из графа G^1 , другой из G^2), которые имеют наибольшую похоть в смысле sim_v . Естественно этим узлам присвоить одинаковое значение l новой метки (в начале алгоритма значение новой метки равно $l = 0$). Далее из списка L удаляется первая тройка, а также все тройки, в которых первый элемент равен v или второй равен $\tilde{v}$; значение l увеличивается на единицу, и процедура присваивания новых меток наиболее похотым узлам повторяется до тех пор, пока список L не пуст. Отметим, что сравниваемые графы могут иметь разное число вершин: $r_1 = |V^1| \neq r_2 = |V^2|$ (здесь и далее запись $|A|$ в случае, когда A – конечное множество, означает количество элементов в этом множестве, а в случае, когда A – число, означает модуль этого числа). В графе, имеющем большее число вершин, некоторые вершины не будут иметь новых меток. Новым меткам этих вершин в произвольном порядке могут быть присвоены значения от $\min\{r_1, r_2\}$ до $\max\{r_1, r_2\} - 1$. Граф с большим числом вершин обозначим G^+ . В граф с меньшим числом вершин, который обозначим G^- , добавляются $\max\{r_1, r_2\} - \min\{r_1, r_2\}$ фиктивных вершин для выравнивания числа вершин в сравниваемых графах. Для каждой из фиктивных вершин значение начальной метки выбирается уникальным (в рамках графа G^-), а новые метки этих вершин принимают значение от $\min\{r_1, r_2\}$ до $\max\{r_1, r_2\} - 1$. Множество добавленных в G^- фиктивных вершин обозначим $F(G^-)$. Будем считать, что ни одна вершина из $F(G^-)$ непохожа ни на одну из вершин графа G^+ :

$$\forall v \in F(G^-), \forall \tilde{v} \in V^+ : \text{sim}_v(v, \tilde{v}) = 0. \quad (2)$$

Пусть $M^1 = (m_{i,j}^1)$ и $M^2 = (m_{i,j}^2)$ – матрицы смежности вершин графов G^1 и G^2 после процедуры назначения новых меток и добавления

фиктивных вершин (если требуется). Отметим, что $m_{i,j}^k = 1$, если в графе G^k из вершины с новой меткой i выходит ребро в вершину с новой меткой j ; в противном случае $m_{i,j}^k = 0$, $k = 1, 2$. Пусть

$$\Delta_{i,j} = |1 - |m_{i,j}^1 - m_{i,j}^2||. \quad (3)$$

Таким образом, $\Delta_{i,j} = 1$, если $m_{i,j}^1 = m_{i,j}^2$, иначе $\Delta_{i,j} = 0$. Результат сравнения графов основан на вычислении нормированной взвешенной разности между матрицами смежности вершин этих графов, которая с учетом (3), здесь определяется следующим образом:

$$\begin{aligned} \rho(G^1, G^2) &= \frac{\sum_{i,j=0}^{\max\{r_1, r_2\}} \Delta_{i,j} \cdot (\text{sim}_v(v_i^1, v_i^2) + \text{sim}_v(v_j^1, v_j^2))}{2 \cdot (\max\{r_1, r_2\})^2} \\ &= \frac{\sum_{i,j=0}^{\min\{r_1, r_2\}} \Delta_{i,j} \cdot (\text{sim}_v(v_i^1, v_i^2) + \text{sim}_v(v_j^1, v_j^2))}{2 \cdot (\max\{r_1, r_2\})^2}, \quad (4) \end{aligned}$$

где v_k^j – начальная метка узла в графе G^j , имеющего новую метку $\text{tag}(v_k^j) = k$, $j = 1, 2$. Равенство (4) вытекает из условия (2). Стоит отметить, что в [21] непохожесть графов предлагалось вычислять путем нахождения нормированного расстояния Хэмминга между матрицами M^1 и M^2 . Другими словами, в [21] похожесть имеет вид:

$$\rho'(G^1, G^2) = \frac{\sum_{i,j=0}^{\max\{r_1, r_2\}} \Delta_{i,j}}{(\max\{r_1, r_2\})^2},$$

где похожесть узлов не учитывается напрямую, а только учитывается при назначении новых меток. Представляется, что формула (4) точнее описывается похожесть сравниваемых графов.

На рисунке 1 в качестве примера изображены два сравниваемых графа $G^1 = (V^1, E^1)$ и $G^2 = (V^2, E^2)$: $V^1 = \{a_1, a_2, a_3, a_4\}$, $V^2 = \{b_1, b_2, b_3, b_4, b_5\}$. В таблице 1 приведены значения *некоторой* функции похожести $\text{sim}_v : V^1 \times V^2 \rightarrow [0, 1]$. В этой таблице жирным шрифтом выделены значения c , соответствующие тройкам $(v, \tilde{v}, c)$, выбираемым на шаге 3 алгоритма sim_G в цикле (шаги с 2 по 7): на первой итерации первой тройкой в списке будет $(b_3, a_1, 0, 8)$, на второй – тройка $(b_1, a_2, 0, 75)$ и т.д.. В этой же таблице серой заливкой отмечены ячейки для фиктивного узла a_5 , который абсолютно непохож ни на один узел графа G^2 :

соответствующие значения функции похожести полагаются равными нулю. Матрицы смежности M^1 и M^2 вершин, построенные на основе новых меток, приведены в таблице 2. Серой заливкой в матрице смежности M^1 выделены столбец и строка для фиктивного узла.

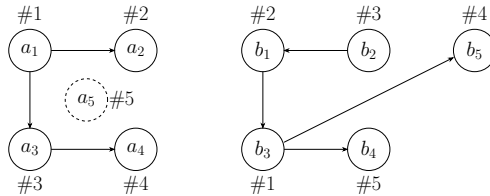


Рис. 1. Пример двух сравниваемых графов G^1 (слева) и G^2 (справа). Начальные метки узлов указаны внутри узлов, а новые метки – рядом с узлами после символа «#». Фиктивный узел, добавленный в граф G^1 , обозначен пунктирной линией

Таблица 1. Значения функции похожести $\text{sim}_v : V^1 \times V^2 \rightarrow [0, 1]$ (пример)

$\text{sim}_v(\cdot, \cdot)$	a_1	a_2	a_3	a_4	a_5
b_1	0,1	0,75	0,23	0,47	0
b_2	0,2	0,1	0,71	0,01	0
b_3	0,8	0,5	0,1	0,4	0
b_4	0,65	0,74	0,62	0,31	0
b_5	0,1	0,6	0,58	0,3	0

Таблица 2. Матрицы смежности вершин графов G^1 и G^2 из примера

M^1	#1	#2	#3	#4	#5	M^2	#1	#2	#3	#4	#5
#1	0	1	1	0	0	#1	0	0	0	1	1
#2	0	0	0	0	0	#2	1	0	0	0	0
#3	0	0	0	1	0	#3	0	1	0	0	0
#4	0	0	0	0	0	#4	0	0	0	0	0
#5	0	0	0	0	0	#5	0	0	0	0	0

Тогда, в соответствии с (4), имеем (в скобках оставлены только ненулевые слагаемые):

$$\begin{aligned}
 \rho(G^1, G^1) &= 50^{-1} \cdot ((0, 8 + 0, 8) + \\
 &+ (0, 75 + 0, 75) + (0, 75 + 0, 71) + (0, 75 + 0, 31) + \\
 &+ (0, 71 + 0, 8) + (0, 71 + 0, 71) + \\
 &+ (0, 31 + 0, 8) + (0, 31 + 0, 75) + (0, 31 + 0, 71) + (0, 31 + 0, 31)) \\
 &= 0, 2472.
 \end{aligned}$$

4. Функция похожести для исполняемых файлов. На основе алгоритма 1 предлагается построение функции похожести для программ.

Именно, программу P представим как направленный граф $G^P = \langle V^P, E^P \rangle$, где $V^P = \{f_1, \dots, f_r\}$ – множество процедур, $E \subseteq V^P \times V^P$ – множество связей между процедурами, причем $(f_i, f_j) \in E$, если из процедуры f_i выполняется обращение к f_j . Такой граф обычно называется графом вызова функций (call function graph). Начальными метками вершин в G^P могут быть, например, названия процедур. Каждую из процедур $f \in V^P$ также можно представить в виде графа $G^f = \langle V^f, E^f \rangle$, где $V^f = \{b_1^f, \dots, b_{n_f}^f\}$ – множество базовых блоков процедур f , то есть таких последовательностей команд, среди которых команда передачи управления находится только в конце последовательности, $E^f \subseteq V^f \times V^f$ – множество связей между базовыми блоками процедуры f . Ребро (b_i^f, b_j^f) принадлежит E^f , если последняя команда блока b_i^f передает управление на блок b_j^f . Граф G^f называется графом потока управления (control flow graph). Начальной меткой каждого узла в G^f может быть, например, адрес первой команды соответствующего базового блока.

Пусть для программ P_1 и P_2 стоит задача оценить их статическую похожесть (без запуска программ). Так как P_1 и P_2 представимы своими графами G^{P_1} и G^{P_2} , сравнение может заключаться в построении подходящей функции похожести sim_f для процедур программ и применении алгоритма 1: $\text{sim}_G(G^{P_1}, G^{P_2}, \text{sim}_f)$. В свою очередь, процедуры программ также представимы графами, узлами в которых являются базовые блоки. Поэтому для сравнения процедур также может быть применён алгоритм 1, если построить подходящую функцию для сравнения базовых блоков. В настоящей работе сравнение базовых блоков предлагается выполнять с помощью хеш-функций $h_i : \{0, 1\}^* \rightarrow \{0, 1\}^{m_i}$, $i = 1, 2$. При этом будем предполагать, что h_1 – криптографическая хеш-функция, а h_2 – нечеткая хеш-функция с алгоритмом сравнения $\text{score} : \{0, 1\}^* \times \{0, 1\}^* \rightarrow [0, 1]$. С каждым базовым блоком $b \in V^f$ процедуры $f \in V^P$ свяжем пару $(h_1(b), h_2(b))$, где $h_i(b)$ – хеш-значение от последовательности ассемблерных команд этого базового блока, $i = 1, 2$. Использование криптографической хеш-функции h_1 позволяет находить идентичные блоки, так как в этом случае хеш-значения совпадают. В случае, когда блоки неидентичны, использовать для сравнения значения криптографической функции не имеет смысла, так как разница аргументов в одном бите приведет в среднем к разнице половины битов значений. Именно поэтому для сравнения неидентичных блоков предлагается использовать нечеткую хеш-функцию h_2 . Функция сравнения sim_{bb} базовых блоков b_1 и b_2 , функция сравнения процедур f_1

и f_2 , функция сравнения программ P_1 и P_2 соответственно имеют вид:

$$\text{sim}_{bb}(b_1, b_2) = \begin{cases} 1, \text{ если } h_1(b_1) = h_1(b_2), \\ \text{score}(h_2(b_1), h_2(b_2)), \text{ иначе,} \end{cases} \quad (5)$$

$$\text{sim}_f(f_1, f_2) = \text{sim}_G(G^{f_1}, G^{f_2}, \text{sim}_{bb}), \quad (6)$$

$$\text{sim}_P(P_1, P_2) = \text{sim}_G(G^{P_1}, G^{P_2}, \text{sim}_f). \quad (7)$$

Далее функцию sim_P будем называть simgraph , подчеркивая, что похожесть вычисляется на основе сравнения графов программ.

В настоящей работе предложенный алгоритм сравнения программ реализован¹ на языке Python с использованием фреймворка дизассемблирования Radare2. В качестве криптографической хеш-функции h_1 и нечеткой хеш-функции h_2 используются соответственно md5 и ssdeer . Выбор в пользу функции md5 с длиной хеш-значения 128 битов обосновывается тем, что число базовых блоков в сравниваемых процедурах существенно меньше $2^{128/2}$, чем обеспечивается пренебрежимая вероятность коллизии (когда два разных базовых блока дают одно и то же хеш-значение). Естественно, что в качестве h_1 может применяться практически любая криптографическая хеш-функция. Более того, от функции h_1 в решаемой задаче требуется только пренебрежимая вероятность коллизии, поэтому h_1 не обязана быть криптографической. Но так как криптографические хеш-функции имеют пренебрежимую вероятность коллизии, то для простоты в функции sim_{bb} (см. формулу (5)) предполагается, что h_1 именно такая. Выбор же функции ssdeer в качестве h_2 обосновывается тем, что эта функция позволяет сравнивать строковые данные произвольной длины, в то время как функция tlsh на вход принимает данные не менее 50 байтов [14], функция mrsh-v2 – не менее 160 байтов, а функция sldhash – не менее 512 байтов [14] [13]. Так как не все базовые блоки имеют длину более 50 байтов, то в качестве h_2 выбрана именно функция ssdeer . Одним из дальнейших направлений исследования является замена функции ssdeer на функцию, вычисляющую расстояние редактирования строк (расстояние Левенштейна), которая применима при сравнении строк любой длины.

5. Результаты экспериментов. В работе проведено два эксперимента (далее – эксперимент 1 и эксперимент 2). В первом эксперименте исследовалось поведение значений построенной функции

¹Реализация размещена на сайте GitHub по адресу <https://github.com/Tempiress/executableFilesComparison>

похожести на множестве пар функционально одинаковых программ, полученных с помощью различных компиляторов и различных опций оптимизации. Ожидается, что для более агрессивных опций оптимизации значения похожести будут меньше, чем для менее агрессивных. Во втором эксперименте исследовалось применение построенной функции похожести в модели оценки эффективности и стойкости обфусцирующих преобразований, предложенной в [10]. Во всех экспериментах для сравнения значений, полученных на основе функции `simgraph`, вычисляется похожесть программ на основе нечетких хеш-функций `ssdeep` и `tlsh`, а также разработанной в [22] функции похожести исполняемых файлов, представленных в виде изображений (далее эта функция называется `simpic`).

5.1. Эксперимент 1. Формирование множества пар функционально одинаковых программ реализовано на основе построенного в [10] множества исполняемых файлов. Именно, множество пар формируется на основе наборов программ `CoreUtils`, `PolyBench` и `HashCat` (всего 164 программы), собранных девятью компиляторами – GCC (версий 7.5.0, 8.4.0, 9.4.0, 10.3.0), Clang (версий 7.0.1, 8.0.1, 9.0.1, 10.0.0) и AOCC (версии 3.0.0) – с пятью опциями оптимизации: O0, O1, O2, O3 и Os. Таким образом, множество исполняемых файлов, построенное в [10], состоит из 7380 файлов. Так как сложность алгоритма $\text{sim}_G - O(|V^1| \times |V^2|)$ (все узлы одного графа сравниваются со всеми узлами другого графа), то для сокращения времени эксперимента из 164-х программ были случайно выбраны 82 программы. Каждая выбранная программа, скомпилированная фиксированным компилятором с опцией O0, сравнивалась с версией той же программы, скомпилированной тем же компилятором, но с опциями O1, O2, O3 и Os. Таким образом, было выполнено $82 \cdot 9 \cdot 4 = 2952$ сравнения с помощью функций `simgraph`, `ssdeep`, `tlsh` и `simpic`. Значения, возвращаемые функциями `ssdeep` и `tlsh`, нормализовались в соответствии с правилами из [15]. Результаты представлены в таблице 3.

Ожидается, что с ростом оптимизации от O1 до O3 значение функции похожести должно падать, так как в рассматриваемых компиляторах опция O2 включает преобразования, подключаемые опцией O1, а опция O3 включает преобразования, задаваемые опцией O2. Сразу отметим, что ни для одной функции эта тенденция не наблюдается на всех рассматриваемых компиляторах. В то же время, для функций `simgraph` и `tlsh` на всех компиляторах программы, полученные с опциями O2 и O3 всегда менее похожи на неоптимизированную программу (O0), чем программа полученная с помощью опции O1. При этом функция `simgraph`

более чувствительна к действию оптимизирующих преобразований, так как отношение похожести в случае (O0,O1) к похожести в случае (O0,O2) или к похожести в случае (O0,O3) выше, чем соответствующие отношения для функции `tlsh`. Для функций `ssdeep` и `simpic` утверждение о том, что «на всех компиляторах программы, полученные с опциями O2 и O3 всегда менее похожи на неоптимизированную программу», не является верным. Именно, для функции `ssdeep` это не выполняется для четырех компиляторов: GCC версий 8.4, 9.4, 10.3 и Clang версии 8.0.1. Для функции `simpic` это не выполняется на компиляторе AOCC и частично на компиляторе Clang версии 9.0.1.

Таблица 3. Усредненные значения функций похожести по 50% программ набора, состоящего из программ пакетов CoreUtils, PolyBench и HashCat

		AOCC	CGG					Clang		
		3.0.0	7.5	8.4	9.4	10.3	7.0.1	8.0.1	9.0.1	10.0.0
simgraph	(O0,O1)	0,087	0,089	0,088	0,088	0,097	0,082	0,082	0,083	0,099
	(O0,O2)	0,067	0,061	0,066	0,066	0,068	0,075	0,075	0,076	0,076
	(O0,O3)	0,067	0,065	0,070	0,071	0,071	0,075	0,075	0,075	0,075
	(O0,O _s)	0,065	0,068	0,063	0,062	0,064	0,074	0,074	0,074	0,074
ssdeep	(O0,O1)	0,162	0,038	0,266	0,271	0,324	0,123	0,115	0,116	0,116
	(O0,O2)	0,135	0,016	0,329	0,375	0,449	0,115	0,136	0,113	0,115
	(O0,O3)	0,127	0,016	0,324	0,403	0,410	0,115	0,133	0,113	0,113
	(O0,O _s)	0,150	0,016	0,459	0,430	0,444	0,116	0,135	0,116	0,116
tlsh	(O0,O1)	0,243	0,55	0,603	0,617	0,613	0,253	0,257	0,257	0,257
	(O0,O2)	0,19	0,517	0,533	0,543	0,527	0,247	0,243	0,243	0,223
	(O0,O3)	0,187	0,4	0,453	0,393	0,42	0,25	0,253	0,253	0,227
	(O0,O _s)	0,137	0,393	0,523	0,537	0,54	0,227	0,203	0,203	0,223
simpic	(O0,O1)	0,634	0,356	0,356	0,355	0,352	0,567	0,555	0,559	0,581
	(O0,O2)	0,640	0,337	0,342	0,339	0,344	0,574	0,544	0,561	0,563
	(O0,O3)	0,675	0,344	0,347	0,350	0,350	0,444	0,463	0,406	0,420
	(O0,O _s)	0,567	0,331	0,331	0,328	0,332	0,580	0,552	0,585	0,600

В целом, разработанная функция `simgraph` демонстрирует наибольшую чувствительность к изменениям, вызванным оптимизацией, по сравнению с функциями `ssdeep`, `tlsh` и `simpic`. Это делает её более подходящей для задач, где требуется более точное сравнение программ на уровне структуры. Рост же похожести в случае (O0,O3) по сравнению со случаем (O0,O2) представляется особенностями компиляторов AOCC и GCC (для всех рассмотренных версий компилятора Clang такой рост не наблюдается), и может являться предметом дальнейших исследований.

5.2. Эксперимент 2. В [10] предложена новая модель оценки эффективности и стойкости обфусцирующих преобразований. Кратко опишем ее. Пусть $\mathcal{O}$ – множество базовых обфусцирующих преобразований. На основе базовых преобразований могут быть

построены последовательности обфусцирующих преобразований. Множество всех (технически реализуемых) последовательностей, которые могут быть построены на основе преобразований из $\mathcal{O}$, в соответствии с [10], обозначим $\mathcal{O}_0^*$. Эффективность e и стойкость r обфусцирующего преобразования $\text{Obf} \in \mathcal{O}_0^*$, примененного к программе P , определены в [10] следующим образом:

$$e(\text{Obf}, P) = 1 - \text{sim}(\text{Obf}(P)) = \text{dis}(\text{Obf}(P), A(P_0)),$$

$$r(D, \text{Obf}, P) = 1 - \text{sim}(D(\text{Obf}(P)), A(P_0)) = \text{dis}(D(\text{Obf}(P)), A(P_0)),$$

где sim – некоторая функция схожести, D – деобфускатор, по отношению к которому оценивается стойкость Obf , A – аппроксимация «самой понятной» версии P_0 программы P . Набор $\mathcal{M} = (\mathcal{O}, \mathcal{Q} \subset \mathcal{O}_0^*, \text{sim}, D, A)$ в [10] назван *моделью оценки эффективности и стойкости обфусцирующих преобразований* из множества $\mathcal{Q}$.

Для реализации этой модели необходимо зафиксировать множество $\mathcal{O}$, выбрать функцию схожести sim исполняемых файлов программ, деобфускатор D и способ аппроксимирования A самой понятной программы P_0 для каждого P . В качестве множества $\mathcal{O}$ в эксперименте 2 выбрано множество преобразований из таблицы 4 (множество преобразований, предоставляемых обфускатором Hikari [23]).

Таблица 4. Набор $\mathcal{O}$ преобразований, предоставляемых компилятором Hikari для программ на языке C [10]

	o	Описание преобразования	Тип
$\mathcal{O}$	bcf	встраивание непрозрачных предикатов	C
	cff	сглаживания графа потока управления	C
	enc	кодирование статических строк	D
	fcw	создание фиктивных функций-прокси	A
	ind	замена инструкций ветвления косвенными переходами	C
	sbb	разбиение базовых блоков	A
	sub	замена инструкций эквивалентными	D
	all	применение всех обфусцирующих преобразований	–

В качестве деобфускатора D использована модель на основе оптимизатора, предложенная и реализованная в [10], а в качестве аппроксимации $A(P_0)$ – предложенная там же самая короткая версия программы, как правило, получаемая с помощью оптимизации с опцией `Os`. В качестве функции sim использованы `simgraph`, `ssdeep`, `tlsh` и `simpic`. В этом эксперименте использованы два набора программ: 20 программ из набора [25] (набор $\mathcal{F}_1$) и 20 случайно выбранных программ из набора, состоящего из пакетов `CoreUtils`, `PolyBench` и `HashCat` (набор $\mathcal{F}_2$).

Заметим, что в последнем столбце таблицы 4 указан тип преобразования, который в [10] определен в соответствии с типами, введенными в [24] для обфускатора Tigress: A – абстрактные преобразования, C – преобразования графа потока управления, D – кодирование данных и арифметических операций. Для каждой последовательности преобразований естественным образом определяется и ее тип. Например, типом последовательности $Obf = (ind, fcw, sbb)$ будет ССА.

В таблицах 5, 6, 7 и 8 представлены типы последовательностей преобразований в упорядоченном по убыванию значений эффективности и стойкости виде.

Таблица 5. Оценка эффективности (e) и стойкости (r) типов обфусцирующих преобразований при использовании функции `simgraph`

e				r			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
DAC	0,865	ACA	0,967	all	0,723	CAA	0,902
DAA	0,837	DAA	0,966	CAA	0,71	DAC	0,889
CAD	0,8215	ACD	0,963	ADA	0,706	DAD	0,885
AC	0,821	ADA	0,963	CCC	0,698	ADA	0,884
ACA	0,813	DAD	0,963	AC	0,697	DAA	0,883
AA	0,811	CAA	0,961	AA	0,691	AC	0,881
CAC	0,809	CAD	0,961	AD	0,691	CAC	0,881
ADA	0,806	AC	0,96	DAA	0,691	ACD	0,874
ACD	0,785	AA	0,958	CCA	0,689	AA	0,872
AD	0,785	AD	0,958	ACA	0,688	CAD	0,871
DAD	0,781	DAC	0,958	ACD	0,688	ACA	0,867
CAA	0,781	CAC	0,951	CAC	0,684	AD	0,856
all	0,724	A	0,865	CAD	0,683	DCA	0,831
A	0,72	DCA	0,861	DAD	0,681	A	0,83
CA	0,719	CA	0,856	DAC	0,676	CA	0,818
DCA	0,717	all	0,843	CA	0,665	DA	0,811
DA	0,693	DA	0,834	DCA	0,665	CCA	0,81
CCA	0,693	CCA	0,821	CDA	0,652	CCC	0,81
CCC	0,673	CC	0,787	DCC	0,65	all	0,801
CDA	0,651	CDA	0,786	CC	0,650	CCD	0,792
CCD	0,638	CCC	0,784	A	0,641	CC	0,789
DCC	0,637	CCD	0,781	DA	0,640	CDA	0,788
CC	0,637	C	0,777	CCD	0,633	C	0,777
DC	0,619	CD	0,776	DC	0,620	CD	0,776
DCD	0,617	DCC	0,775	DCD	0,619	DCC	0,776
CD	0,615	DC	0,770	DD	0,615	DC	0,770
C	0,614	DCD	0,768	DDA	0,615	DCD	0,770
DD	0,614	DDA	0,768	CD	0,614	D	0,769
D	0,614	D	0,768	C	0,613	DDA	0,768
DDA	0,611	DD	0,766	D	0,6125	DD	0,767

Из таблицы 6 видно, что при использовании функции *ssdeep* последовательности мало отличаются, особенно на наборе $\mathcal{F}_2$. Отсюда можно заключить, что эта функция малопригодна в задаче оценки эффективности и стойкости обфусцирующих преобразований.

Таблица 6. Оценка эффективности (e) и стойкости (r) типов обфусцирующих преобразований при использовании функции *ssdeep*

e				r			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
all	1, 0	A	1, 0	all	1, 0	A	1, 0
CAA	1, 0	AA	1, 0	DAA	1, 0	AA	1, 0
DDA	1, 0	DDA	1, 0	CCC	1, 0	DDA	1, 0
DCA	1, 0	DD	1, 0	DCC	0, 994	DD	1, 0
CCC	1, 0	DCD	1, 0	CCA	0, 993	DCD	1, 0
CCA	1, 0	DCC	1, 0	DC	0, 991	DCC	1, 0
DCC	0, 994	DCA	1, 0	DCA	0, 988	DCA	1, 0
DCD	0, 992	DC	1, 0	CAA	0, 986	DC	1, 0
CDA	0, 991	DAD	1, 0	DCD	0, 984	DAD	1, 0
CCD	0, 99	DAC	1, 0	CC	0, 983	DAC	1, 0
CAC	0, 987	DAA	1, 0	CDA	0, 983	DAA	1, 0
CAD	0, 986	DA	1, 0	DDA	0, 98	DA	1, 0
CA	0, 986	D	1, 0	CCD	0, 98	D	1, 0
DC	0, 983	CDA	1, 0	CA	0, 979	CDA	1, 0
CC	0, 983	CD	1, 0	ADA	0, 978	CD	1, 0
CD	0, 981	CCD	1, 0	DD	0, 977	CCD	1, 0
DD	0, 977	CCC	1, 0	A	0, 976	CCC	1, 0
AD	0, 976	CCA	1, 0	DAD	0, 976	CCA	1, 0
DAD	0, 976	CC	1, 0	AA	0, 976	CC	1, 0
A	0, 976	CAD	1, 0	AD	0, 975	CAD	1, 0
ADA	0, 976	CAC	1, 0	D	0, 974	CAC	1, 0
DA	0, 975	CAA	1, 0	CAD	0, 974	CAA	1, 0
AA	0, 975	CA	1, 0	C	0, 973	CA	1, 0
DAA	0, 975	C	1, 0	DA	0, 973	C	1, 0
DAC	0, 974	ADA	1, 0	ACD	0, 973	ADA	1, 0
ACA	0, 974	AD	1, 0	DAC	0, 973	AD	1, 0
AC	0, 974	ACD	1, 0	ACA	0, 972	ACD	1, 0
D	0, 974	ACA	1, 0	CD	0, 972	ACA	1, 0
C	0, 973	AC	1, 0	CAC	0, 972	AC	1, 0
ACD	0, 972	all	1, 0	AC	0, 966	all	1, 0

Эти результаты коррелируют с результатами эксперимента 1 и результатами работы [11], где в рамках другого эксперимента выявлен незначительный вес признака, вычисленного с использованием функции *ssdeep*. Стоит отметить, что низкая точность результатов функции *ssdeep* впервые была продемонстрирована в [15]. Поэтому далее из рассмотрения функция *ssdeep* исключена. Здесь стоит отметить, что этот вывод не

противоречит использованию функции *ssdeep* в реализации функции *sim_{bb}*: в *sim_{bb}* функция *ssdeep* используется только для сравнения базовых блоков, а сделанный вывод относится к случаю, когда *ssdeep* используется для сравнения файлов полностью.

Таблица 7. Оценка эффективности (e) и стойкости (r) типов обфусцирующих преобразований при использовании функции *tlsh*

e				r			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
all	0,998	all	1,0	all	0,983	all	1,0
CCC	0,963	CCC	1,0	DCC	0,894	CCC	0,989
DCC	0,939	DDA	0,993	DCA	0,877	DCA	0,946
CCA	0,923	DAD	0,991	DAC	0,861	DDA	0,946
DCD	0,917	DCD	0,991	CCA	0,86	DAD	0,941
DCA	0,912	DCC	0,988	DAA	0,858	DCC	0,94
CCD	0,897	CCA	0,979	DCD	0,855	DCD	0,938
DC	0,89	DCA	0,971	DC	0,853	DAA	0,935
DDA	0,889	DD	0,968	DAD	0,841	CCA	0,934
DAD	0,888	CAA	0,966	CCC	0,834	DD	0,909
CAA	0,888	DC	0,931	DD	0,825	CAA	0,906
DAC	0,872	DAA	0,93	CAA	0,818	DC	0,895
CC	0,871	CC	0,916	DDA	0,809	DAC	0,883
DD	0,864	CCD	0,913	AC	0,808	DA	0,854
DAA	0,863	CDA	0,895	ACD	0,794	CDA	0,846
CAD	0,846	CAC	0,89	D	0,786	CC	0,817
CAC	0,842	CA	0,887	DA	0,781	CCD	0,814
CA	0,826	DAC	0,873	CC	0,775	AA	0,81
CDA	0,814	CAD	0,872	CAC	0,772	ADA	0,795
AC	0,804	DA	0,852	ACA	0,769	CA	0,793
CD	0,8	C	0,805	CDA	0,768	ACA	0,787
DA	0,797	CD	0,804	AD	0,768	D	0,771
ACD	0,79	ADA	0,798	CA	0,766	CAD	0,752
AD	0,79	AD	0,783	A	0,754	CAC	0,752
C	0,781	AA	0,773	AA	0,749	A	0,748
D	0,776	ACA	0,769	CCD	0,745	AC	0,719
A	0,758	D	0,75	ADA	0,737	CD	0,716
AA	0,741	A	0,724	CAD	0,736	C	0,7
ADA	0,735	ACD	0,722	C	0,725	ACD	0,69
ACA	0,732	AC	0,704	CD	0,72	AD	0,679

В таблице 9 (а, б, в) результаты из таблиц 5, 7 и 8 сгруппированы по последовательностям длины 3, 2 и 1 соответственно. Из таблиц видно, что для *simgraph* последовательности обфусцирующих преобразований, содержащие базовые преобразования типа А, обладают более высокой эффективностью и стойкостью (эти последовательности сосредоточены ближе к верху таблицы).

Таблица 8. Оценка эффективности (e) и стойкости (r) типов обфусцирующих преобразований при использовании функции `simprc`

e				r			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
all	0, 406	all	0, 791	all	0, 365	all	0, 784
DCC	0, 247	CCA	0, 756	DAC	0, 257	CCA	0, 734
CDA	0, 241	AA	0, 745	CCD	0, 231	DAC	0, 729
CCA	0, 239	CCC	0, 739	CAA	0, 220	DCC	0, 726
CAD	0, 235	ADA	0, 727	CAC	0, 215	CCC	0, 716
CCD	0, 225	CAA	0, 726	CDA	0, 215	CCD	0, 714
DCD	0, 225	CDA	0, 718	D	0, 212	AC	0, 710
DAA	0, 223	DAD	0, 713	A	0, 211	CC	0, 707
ACD	0, 222	ACA	0, 690	CCA	0, 210	ACD	0, 707
DA	0, 221	CCD	0, 684	ADA	0, 206	AD	0, 707
A	0, 220	CA	0, 683	DD	0, 204	DCD	0, 706
DCA	0, 217	DD	0, 672	DAA	0, 202	DCA	0, 705
DDA	0, 216	CD	0, 668	AA	0, 200	C	0, 703
CAC	0, 216	CC	0, 664	CAD	0, 189	ADA	0, 703
ACA	0, 210	A	0, 661	ACA	0, 188	CAA	0, 702
AA	0, 208	C	0, 661	ACD	0, 187	ACA	0, 701
AC	0, 203	CAC	0, 661	DCA	0, 186	A	0, 700
CA	0, 202	ACD	0, 649	C	0, 185	CDA	0, 699
ADA	0, 202	AC	0, 644	AD	0, 182	CAD	0, 696
DD	0, 199	DCD	0, 634	DDA	0, 179	CD	0, 693
CAA	0, 196	CAD	0, 632	CA	0, 177	CA	0, 692
CCC	0, 195	AD	0, 619	CD	0, 177	D	0, 692
CC	0, 191	D	0, 590	DCC	0, 175	DA	0, 692
DAC	0, 185	DDA	0, 571	DCD	0, 174	AA	0, 690
DC	0, 185	DA	0, 541	CCC	0, 171	DAD	0, 687
C	0, 181	DAC	0, 523	CC	0, 169	DD	0, 685
AD	0, 176	DCC	0, 516	DA	0, 166	DC	0, 680
DAD	0, 175	DCA	0, 489	DC	0, 162	DAA	0, 679
CD	0, 161	DAA	0, 471	AC	0, 162	DDA	0, 678
D	0, 140	DC	0, 454	DAD	0, 127	CAC	0, 666

Такой результат объясняется тем, что преобразования типа А, согласно таблице 4, меняют как граф вызова функций/процедур (добавляются фиктивные функции), так и граф потока управления (разбиваются базовые блоки). Для последовательностей преобразований длины один на разных наборах функция `simgraph` одинаково ранжировала типы как в рамках оценки эффективности, так и в рамках оценки стойкости. При этом как наиболее эффективными, так и наиболее стойкими являются одиночные преобразования типа А. Примечательно, что преобразование типа DDA на обоих наборах демонстрирует наименьшую эффективность и стойкость (таблица 9(а)).

Таблица 9. Оценка эффективности (e) и стойкости (r) типов обфусцирующих преобразований длины: а) 3; б) 2; в) 1

а)											
simgraph				tlsh				simpic			
e		r		e		r		e		r	
$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$
DAC	ACA	CAA	CAA	CCC	CCC	DCC	CCC	DCC	CCA	DAC	CCA
DAA	DAA	ADA	DAC	DCC	DDA	DCA	DCA	CDA	CCC	CCD	DAC
CAD	ACD	CCC	DAD	CCA	DAD	DAC	DDA	CCA	ADA	CAA	DCC
ACA	ADA	DAA	ADA	DCD	DCD	CCA	DAD	CAD	CAA	CAC	CCC
CAC	DAD	CCA	DAA	DCA	DCC	DAA	DCC	CCD	CDA	CDA	CCD
ADA	CAA	ACA	CAC	CCD	CCA	DCD	DCD	DCD	DAD	CCA	ACD
ACD	CAD	ACD	ACD	DDA	DCA	DAD	DAA	DAA	ACA	ADA	DCD
DAD	DAC	CAC	CAD	DAD	CAA	CCC	CCA	ACD	CCD	DAA	DCA
CAA	CAC	CAD	ACA	CAA	DAA	CAA	CAA	DCA	CAC	CAD	ADA
DCA	DCA	DAD	DCA	DAC	CCD	DDA	DAC	DDA	ACD	ACA	CAA
CCA	CCA	DAC	CCA	DAA	CDA	ACD	CDA	CAC	DCD	ACD	ACA
CCC	CDA	DCA	CCC	CAD	CAC	CAC	CCD	ACA	CAD	DCA	CDA
CDA	CCC	CDA	CCD	CAC	DAC	ACA	ADA	ADA	DDA	DDA	CAD
CCD	CCD	DCC	CDA	CDA	CAD	CDA	ACA	CAA	DAC	DCC	DAD
DCC	DCC	CCD	DCC	ACD	ADA	CCD	CAD	CCC	DCC	DCD	DAA
DCD	DCD	DCD	DCD	ADA	ACA	ADA	CAC	DAC	DCA	CCC	DDA
DDA	DDA	DDA	DDA	ACA	ACD	CAD	ACD	DAD	DAA	DAD	CAC

б)											
simgraph				tlsh				simpic			
e		r		e		r		e		r	
$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$
AC	AC	AC	AC	DC	DD	DC	DD	DA	AA	DD	AC
AA	AA	AA	AA	CC	DC	DD	DC	AA	CA	AA	CC
AD	AD	AD	AD	DD	CC	AC	DA	AC	DD	AD	AD
CA	CA	CA	CA	CA	CA	DA	CC	CA	CD	CA	CD
DA	DA	CC	DA	AC	DA	CC	AA	DD	CC	CD	CA
CC	CC	DA	CC	CD	CD	AD	CA	CC	AC	CC	DA
DC	CD	DC	CD	DA	AD	CA	AC	DC	AD	DA	AA
CD	DC	DD	DC	AD	AA	AA	CD	AD	DA	DC	DD
DD	DD	CD	DD	AA	AC	CD	AD	CD	DC	AC	DC

в)											
simgraph				tlsh				simpic			
e		r		e		r		e		r	
$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_1$	$\mathcal{F}_2$
A	A	A	A	C	C	D	D	A	A	D	C
C	C	C	C	D	D	A	A	C	C	A	A
D	D	D	D	A	A	C	C	D	D	C	D

Для функций tlsh большей эффективностью и стойкостью обладают последовательности, содержащие преобразования типа D, C и их комбинации. При этом преобразования типа A располагаются

ближе к низу таблицы. Для одиночных преобразований наиболее эффективными являются преобразования типа C, а наиболее стойкими – преобразования типа D. Наблюдаются одинаковые результаты для разных наборов программ, однако для эффективности и стойкости результаты оценки не совпадают. Для функции *simpic* сложнее выделить базовые преобразования, сочетания которых приводят к наиболее эффективным и стойким преобразованиям. И для одиночных преобразований эта функция показывает разные результаты относительно стойкости для разных наборов программ, при этом для эффективности результаты одинаковые и повторяют результат для функции *simgraph*.

Таким образом, функции *simgraph* и *tlsh* ведут себя стабильно на разных наборах программ. Однако для *tlsh* преобразования типа C считаются наименее стойкими, что не соответствует эмпирическим результатам и оценкам, полученным с помощью других функций [10].

В таблице 10 указаны типы последовательностей, которые для обоих наборов $\mathcal{F}_1$ и $\mathcal{F}_2$ попали в группу из лучших (худших) типов последовательностей, имеющих наибольшее (соответственно наименьшее) значение эффективности, стойкости или и того, и другого одновременно. Из этой таблицы видно, что для всех трех функций: 1) преобразование типа DAA входит в 15 наиболее эффективных преобразований, 2) преобразования типа CAA и DAC входят в 15 наиболее стойких преобразований, 3) преобразование типа D входит в 15 наименее эффективных, 4) преобразование типа CD входит в 15 наименее стойких. Стоит отметить, что эти результаты частично коррелируют с результатами работы [10], где эффективность и стойкость оценивалась с помощью функции похожести, построенной в [11] методами машинного обучения на базе набора специализированных и универсальных характеристик (среди которых имеются характеристики на основе нечетких хеш-функций). Построенная в [11] функция позволила в [10] выделить наиболее стойкие и наиболее слабые типы преобразований, которые таковыми являются, исходя из результатов других независимых работ. Результаты настоящей работы коррелируют с результатами работы [10] в том, что одиночные преобразования типа D считаются наименее эффективными, а преобразования типа CAA относятся к числу наиболее стойких.

Таблица 10. Типы обфусцирующих последовательностей, входящие в N лучших ($T(N)$) и худших ($B(N)$) типов последовательностей, как для $\mathcal{F}_1$, так и для $\mathcal{F}_2$

	e	r	e, r
simgraph	T3	DAA	ADA, CAA
	B3	D	—
	T7	CAD, DAA, ACA, AC, ADA	ADA, CAA, AC, DAA
	B7	DCD, DDA, DD, D	DAA, AC, ADA
	T15	all, CAD, DAA, ACD, A, DCA, CAC, ACA, AA, AC, CA, ADA, CAA, DAC, AD, DAD	DCD, DDA, DD, D CAC, ACA, AA, AC, ADA, DAA, ACD, DCA, CAD, CAA, DAC, AD, DAD
	B15	CD, D, DC, C, DD, DDA, DCC, CDA, CCA, CCD, DCD, DA, CCC, CC	DCD, DDA, DD, D DCC, CDA, CCD, DCD, DDA, DD, CC, DC, C, CD, D
tlsh	T3	all, CCC	all, DCA
	B3	—	—
	T7	all, DCC, CCA, DCD, DCA, CCC	all, DCA, DCC, DCD, DAA
	B7	A, ACA, AA, D	C, CD
	T15	all, DCC, CCA, CCD, DCD, DAA, DCA, DDA, DD, CAA, CCC, CC, DC, DAD	all, DCC, CCA, DCD, DAA, DCA, DDA, DD, CAA, CCC, DAC, DC, DAD
	B15	ACD, DA, A, ACA, AA, AC, CA, ADA, C, AD, CD, D	CAD, CCD, A, CAC, ACA, AA, CA, ADA, C, AD, CD
simpic	T3	all, CCA	all, DAC
	B3	—	—
	T7	all, CDA, CCA	all, CCD, DAC
	B7	DC	DC, DAD
	T15	all, CDA, CCA, CCD, A, DAA, ACA, AA	all, CCA, CCD, ACD, ACA, ADA, CAA, DAC
	B15	AC, DAC, DC, AD, D	DA, DDA, CA, DC, DAD, CD

6. Заключение. Разработка новых алгоритмов сравнения бинарных исполняемых файлов является актуальной задачей, так как новые функции схожести находят применение в разных областях программной инженерии. Полученные в настоящей работе результаты можно разделить на две части.

К первой части результатов, с одной стороны, относится новый способ сравнения бинарных исполняемых файлов, описываемый формулами (5)–(7), а с другой стороны, потенциальные возможности построения новых функций схожести на базе этого способа. В предлагаемом в настоящей работе способе используется общий для

многих специализированных алгоритмов подход [16–20], включающий сравнение графов процедур и сравнение графов программ. Новым в предлагаемом способе является использование нового алгоритма сравнения графов sim_G как для сравнения графов процедур в функции sim_f , так и для сравнения графов программ в функции sim_P . Алгоритм sim_G может быть основой и для построения других алгоритмов сравнения. Например, такие алгоритмы можно получить, заменив в функции sim_P ((7)) функцию сравнения процедур sim_f , например, алгоритмами сравнения процедур из [16–20]. И наоборот, в известных алгоритмах сравнения графов программ алгоритм сравнения процедур может быть заменен на sim_f ((6)). Кроме того, в самой функции sim_f возможно замена sim_{bb} другой функцией сравнения базовых блоков. Перспективным представляется использование функции сравнения базовых блоков вида

$$\widetilde{\text{sim}}_{bb}(b_1, b_2, T) = \begin{cases} 1, & \text{если } h_1(b_1) = h_1(b_2), \\ \text{score}(h_2(T(b_1)), h_2(T(b_2))), & \text{иначе,} \end{cases}$$

где алгоритм T выполняет преобразование инструкций базового блока перед вычислением хеш-значения. Например, алгоритм T может заменять ассемблерные команды типом команды (арифметическая операция, пересылка данных и т.п.) и/или заменять конкретные аргументы команд их кодами (регистр, ячейка памяти, константа и т.п.). Другие функции сравнения могут быть также получены даже в рамках способа сравнения, описанного формулами (5)–(7), например, путем использования другого фреймворка дизассемблирования или использования других нечетких хеш-функций при сравнении базовых блоков процедур.

Среди обозначенного потенциального разнообразия вариантов сравнения бинарных исполняемых файлов в работе реализован способ, описанный формулами (5)–(7), с использованием фреймворка дизассемблирования Radare2, криптографической функции md5 и нечеткой хеш-функции ssdeep. Реализация этого способа и проведение экспериментов составляют вторую часть результатов настоящей работы. Проведенные эксперименты показали, что, благодаря повышенной чувствительности к оптимизационным изменениям, функция simgraph превосходит ssdeep , tlsh и simpic в точности сравнения структурного содержания программ, что определяет её преимущество для соответствующих задач. Более высокая чувствительность проявляется в том, что для функции simgraph отношение похожести в случае

(00,01) к похожести в случае (00,02) или к похожести в случае (00,03) всегда выше, чем соответствующие отношения у других функций (см. таблицу 3). Результаты использования функции *simgraph* в модели оценки эффективности и стойкости обфусцирующих преобразований показали, что построенная функция вряд ли может самостоятельно использоваться для такой оценки, так как, в силу устройства алгоритма *sim_G*, наиболее эффективными и стойкими считаются обфусцирующие последовательности, включающие преобразования типа А. Это расходится с эмпирическими результатами независимых исследований, в которых наиболее эффективными и стойкими считаются обфусцирующие последовательности с преобразованиями типа С. Тем не менее, функция *simgraph* может применяться для оценки степени запутанности графа выполнения программ, так как сравнение выполняется на основе графа. Ее достоинством представляется возможность интерпретации значений этой функции: меньшие значения соответствуют меньшей похожести на уровне графов программ и базовых блоков, и наоборот. Представляется, что *simgraph* может использоваться для построения более точных функций похожести, например, на основе методов машинного обучения, как это сделано в [11].

Стоит отметить высокую сложность разработанного алгоритма сравнения графов, так как на этапе сопоставления вершин графов все вершины одного графа сравниваются со всеми вершинами другого графа. Одним из дальнейших направлений исследования является поиск путей уменьшения сложности сравнения.

Авторы благодарят рецензентов за замечания и соответствующие предложения, позволившие лучше представить полученные результаты.

Литература

1. Van Tilborg H., Jajodia S. Encyclopedia of cryptography and security. Springer Science & Business Media. 2014. 1416 p.
2. Li B., He J., Huang J., Shi Y.Q. A survey on image steganography and steganalysis // Journal of Information Hiding and Multimedia Signal Processing. 2011. vol. 2. no. 2. pp. 142–172.
3. Chabot C. Recognition of a code in a noisy environment // IEEE International Symposium on Information Theory. IEEE, 2007. pp. 2211–2215. DOI: 10.1109/ISIT.2007.4557548.
4. Crawford M., Khoshgoftaar T.M., Prusa J.D., Richter A.N., Al Najada H. Survey of review spam detection using machine learning techniques // Journal of Big Data. 2015. vol. 2. DOI: 10.1186/s40537-015-0029-9.
5. Forrest S., Hofmeyr S., Somayaji A. The evolution of system-call monitoring // Proceedings of the 2008 Annual Computer Security Applications Conference (ACSAC). 2008. pp. 418–430. DOI: 10.1109/ACSAC.2008.5.
6. Khraisat A., Gondal I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges // Cybersecurity. 2019. vol. 2. DOI: 10.1186/s42400-019-0038-7.

7. Kosolapov Y.V. On detecting code reuse attacks // *Automatic Control and Computer Sciences*. 2020. vol. 54. no. 7. pp. 573–583. DOI: 10.3103/S0146411620070111.
8. Kiger J., Ho S.-S., Heydari V. Malware binary image classification using convolutional neural networks // *Proceedings of the 17th International Conference on Cyber Warfare and Security (ICWS)*. 2022. vol. 17. pp. 469–478. DOI: 10.34190/icws.17.1.59.
9. Polsani H., Jiang H., Liu Y. DeepGray: Malware Classification Using Grayscale Images with Deep Learning // *The 37th International FLAIRS Conference*. 2024. pp. 1–5.
10. Борисов П.Д., Косолапов Ю.В. Способ количественного сравнения обфусцирующих преобразований // *Информатика и автоматизация*. 2024. Т. 23. № 3. С. 684–726. DOI: 10.15622/ia.23.3.3.
11. Борисов П.Д., Косолапов Ю.В. Способ оценки похожести программ методами машинного обучения // *Труды Института системного программирования РАН*. 2022. Т. 34. № 5. С. 63–76. DOI: 10.15514/ISPRAS-2022-34(5)-4.
12. Kornblum J. Identifying almost identical files using context triggered piecewise hashing // *Digital investigation*. 2006. vol. 3. pp. 91–97. DOI: 10.1016/j.diin.2006.06.015.
13. Breiting F., Baier H. Similarity preserving hashing: Eligible properties and a new algorithm mrsh-v2 // *Digital Forensics and Cyber Crime: 4th International Conference (ICDF2C 2012)*. 2013. pp. 167–182. DOI: 10.1007/978-3-642-39891-9_11.
14. Roussev V. An evaluation of forensic similarity hashes // *Digital investigation*. 2011. vol. 8. pp. S34–S41. DOI: 10.1016/j.diin.2011.05.005.
15. Pagani F., Dell’Amico M., Balzarotti D. Beyond precision and recall: understanding uses (and misuses) of similarity hashes in binary analysis // *Proc. of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018. pp. 354–365. DOI: 10.1145/3176258.3176306.
16. BinDiff. URL: <https://www.zynamics.com/> (дата обращения: 23.06.2025).
17. Aslanyan H., Avetisyan A., Arutunian M., Keropyan G., Kurmangaleev S., Vardanyan V. Scalable Framework for Accurate Binary Code Comparison // *Ivannikov ISPRAS Open Conference (ISPRAS)*. 2017. pp. 34–38. DOI: 10.1109/ISPRAS.2017.00013.
18. Machoc hash. URL: <https://github.com/ANSSI-FR/polichombr/blob/dev/> (дата обращения: 23.06.2025).
19. Machoke. URL: <https://github.com/conix-security/machoke> (дата обращения: 23.06.2025).
20. Li Y., Jang J., Ou X. Topology-aware hashing for effective control flow graph similarity analysis // *Security and Privacy in Communication Networks: 15th EAI International Conference (SecureComm)*. 2019. pp. 278–298.
21. Borisov P.D., Kosolapov Y.V. On the Automatic Analysis of the Practical Resistance of Obfuscating Transformations // *Aut. Control Comp. Sci.* 2020. vol. 54. pp. 619–629. DOI: 10.3103/S0146411620070044.
22. Борисов П.Д., Косолапов Ю.В. О функции похожести графических представлений исполняемых файлов в модели оценки обфусцирующих преобразований // *Известия ЮФУ*. 2025. № 3(245). С. 264–273.
23. Naville Z. Hikari—an improvement over Obfuscator-LLVM. 2017. URL: <https://github.com/HikariObfuscator/Hikari> (дата обращения: 26.11.2024).
24. Holder W., McDonald J.T., Andel T.R. Evaluating optimal phase ordering in obfuscation executives // *Proceedings of the 7th Software Security, Protection, and Reverse Engineering/Software Security and Protection Workshop*. 2017. pp. 1–12. DOI: 10.1145/3151137.3151140.
25. small-programs. A set of small programs for experiments with obfuscations. URL: <https://github.com/Boriskin61/small-programs> (дата обращения: 22.06.2025).

Борисов Петр Дмитриевич — заведующий лабораторией, ФГАНУ НИИ «Спецвузавтоматика». Область научных интересов: исследование и анализ программного кода, методы обфускации и деобфускации, способы оценки качества обфусцирующих преобразований. Число научных публикаций — 10. borisovpetr@mail.ru; улица Города Волос, 6, 344011, Ростов-на-Дону, Россия; р.т.: +7(863)297-5111.

Варламов Данил Викторович — студент, кафедра алгебры и дискретной математики института математики, механики и компьютерных наук им. и.и. воровича, Южный федеральный университет (ЮФУ). Область научных интересов: исследование и анализ программного кода. varlamov@sfnedu.ru; улица Мильчакова, 8а, 344090, Ростов-на-Дону, Россия; р.т.: +7(863)297-5111.

Косолапов Юрий Владимирович — канд. техн. наук, доцент кафедры, кафедра алгебры и дискретной математики института математики, механики и компьютерных наук им. и.и. воровича, Южный федеральный университет (ЮФУ). Область научных интересов: помехоустойчивые коды в криптографии и стеганографии, теоретическая и практическая обфускация кода. Число научных публикаций — 110. yvkosolapov@sfnedu.ru; улица Мильчакова, 8а, 344090, Ростов-на-Дону, Россия; р.т.: +7(863)297-5111.

P. BORISOV, D. VARLAMOV, Yu. KOSOLAPOV
**A GRAPH SIMILARITY CALCULATION ALGORITHM AND ITS
APPLICATION FOR COMPARING BINARY EXECUTABLE FILES**

Borisov P., Varlamov D., Kosolapov Yu. A Graph Similarity Calculation Algorithm and Its Application for Comparing Binary Executable Files.

Abstract. The paper addresses the task of static (without execution) comparison of binary executable files. A program and any of its procedures can be represented as a directed graph. For a program, the corresponding graph is a function (procedure) call graph, where the nodes are the functions themselves, and an edge from vertex a to b describes a call to function b from function a . For a procedure, such a graph is a control flow graph, where the vertices are basic blocks, and an edge between nodes a and b means that the commands of block b can be executed after the commands of block a . The study proposes an algorithm for comparing directed graphs, which is then applied to program comparison. The graph comparison algorithm is based on applying a node similarity function. For comparing procedure graphs, a fuzzy hash function and a cryptographic hash function are used as this similarity function. Subsequently, this method of comparing procedure graphs is used as the node similarity function for comparing program graphs. Based on the proposed algorithm, a method for program comparison has been developed, and its investigation was conducted through two experiments. In the first experiment, the method's behavior was studied when comparing programs compiled with different optimization options (O0, O1, O2, O3, and Os). In the second experiment, the possibility of identifying effective and resilient obfuscating transformations within a previously developed model was investigated. Within the first experiment, evidence was obtained supporting the hypothesis that similarity decreases as optimization increases from O1 to O3. Within the second experiment, some previously obtained results concerning the effectiveness (ineffectiveness) and resilience (non-resilience) of obfuscating transformations were confirmed.

Keywords: graph comparison, program similarity, effectiveness and resilience of obfuscation.

References

1. Van Tilborg H., Jajodia S. Encyclopedia of cryptography and security. Springer Science & Business Media. 2014. 1416 p.
2. Li B., He J., Huang J., Shi Y.Q. A survey on image steganography and steganalysis. Journal of Information Hiding and Multimedia Signal Processing. 2011. vol. 2. no. 2. pp. 142–172.
3. Chabot C. Recognition of a code in a noisy environment. IEEE International Symposium on Information Theory. IEEE, 2007. pp. 2211–2215. DOI: 10.1109/ISIT.2007.4557548.
4. Crawford M., Khoshgoftaar T.M., Prusa J.D., Richter A.N., Al Najada H. Survey of review spam detection using machine learning techniques. Journal of Big Data. 2015. vol. 2. DOI: 10.1186/s40537-015-0029-9.
5. Forrest S., Hofmeyr S., Somayaji A. The evolution of system-call monitoring. Proceedings of the 2008 Annual Computer Security Applications Conference (ACSAC). 2008. pp. 418–430. DOI: 10.1109/ACSAC.2008.5.
6. Khraisat A., Gondal I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity. 2019. vol. 2. DOI: 10.1186/s42400-019-0038-7.

7. Kosolapov Y.V. On detecting code reuse attacks. *Automatic Control and Computer Sciences*. 2020. vol. 54. no. 7. pp. 573–583. DOI: 10.3103/S0146411620070111.
8. Kiger J., Ho S.-S., Heydari V. Malware binary image classification using convolutional neural networks. *Proceedings of the 17th International Conference on Cyber Warfare and Security (ICCWS)*. 2022. vol. 17. pp. 469–478. DOI: 10.34190/iccws.17.1.59.
9. Polsani H., Jiang H., Liu Y. DeepGray: Malware Classification Using Grayscale Images with Deep Learning. *The 37th International FLAIRS Conference*. 2024. pp. 1–5.
10. Borisov P.D., Kosolapov Yu.V. [A method to quantitative compare obfuscating transformations]. *Informatics and Automation*. 2024. vol. 23. no. 3. pp. 684–726. DOI: 10.15622/ia.23.3.3. (In Russ.).
11. Borisov P.D., Kosolapov Yu.V. [Method to Evaluate Program Similarity Using Machine Learning Methods]. *Trudy Instituta sistemnogo programmirovaniya RAN – Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2022. vol. 34. no. 5. pp. 63–76. DOI: 10.15514/ISPRAS-2022-34(5)-4. (In Russ.).
12. Kornblum J. Identifying almost identical files using context triggered piecewise hashing. *Digital investigation*. 2006. vol. 3. pp. 91–97. DOI: 10.1016/j.diin.2006.06.015.
13. Breitinger F., Baier H. Similarity preserving hashing: Eligible properties and a new algorithm mrsh-v2. *Digital Forensics and Cyber Crime: 4th International Conference (ICDF2C 2012)*. 2013. pp. 167–182. DOI: 10.1007/978-3-642-39891-9_11.
14. Roussev V. An evaluation of forensic similarity hashes. *Digital investigation*. 2011. vol. 8. pp. S34–S41. DOI: 10.1016/j.diin.2011.05.005.
15. Pagani F., Dell'Amico M., Balzarotti D. Beyond precision and recall: understanding uses (and misuses) of similarity hashes in binary analysis. *Proc. of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018. pp. 354–365. DOI: 10.1145/3176258.3176306.
16. BinDiff. Available at: <https://www.zynamics.com/> (accessed 23.06.2025).
17. Aslanyan H., Avetisyan A., Arutunian M., Keropyan G., Kurmangaleev S., Vardanyan V. Scalable Framework for Accurate Binary Code Comparison. *Ivannikov ISPRAS Open Conference (ISPRAS)*. 2017. pp. 34–38. DOI: 10.1109/ISPRAS.2017.00013.
18. Machoc hash. Available at: <https://github.com/ANSSI-FR/polichombr/blob/dev/> (accessed 23.06.2025).
19. Machoke. Available at: <https://github.com/conix-security/machoke> (accessed 23.06.2025).
20. Li Y., Jang J., Ou X. Topology-aware hashing for effective control flow graph similarity analysis. *Security and Privacy in Communication Networks: 15th EAI International Conference (SecureComm)*. 2019. pp. 278–298.
21. Borisov P.D., Kosolapov Y.V. On the Automatic Analysis of the Practical Resistance of Obfuscating Transformations. *Aut. Control Comp. Sci.* 2020. vol. 54. pp. 619–629. DOI: 10.3103/S0146411620070044.
22. Borisov P.D., Kosolapov Yu.V. [On the Similarity Function of Graphical Representations of Executable Files in the Obfuscation Transformation Evaluation Model]. *Izvestiya JuFU – Bulletin of the Southern Federal University*. 2025. no. 3(245). pp. 264–273. (In Russ.).
23. Naville Z. Hikari—an improvement over Obfuscator-LLVM. 2017. Available at: <https://github.com/HikariObfuscator/Hikari> (accessed 26.11.2024).
24. Holder W., McDonald J.T., Andel T.R. Evaluating optimal phase ordering in obfuscation executives. *Proceedings of the 7th Software Security, Protection, and Reverse Engineering/Software Security and Protection Workshop*. 2017. pp. 1–12. DOI: 10.1145/3151137.3151140.
25. small-programs. A set of small programs for experiments with obfuscations. Available at: <https://github.com/Boriskin61/small-programs> (accessed 22.06.2025).

Borisov Petr — Head of the laboratory, FSASE SRI «Specvuzavtomatika». Research interests: program code analysis, obfuscation and deobfuscation, methods for assessing the quality of obfuscating transformations. The number of publications — 10. borisovpetr@mail.ru; 6, Goroda Volos St., 344011, Rostov-on-Don, Russia; office phone: +7(863)297-5111.

Varlamov Danil — Student, Department of algebra and discrete mathematics, i.i. vorovich institute of mathematics, mechanics, and computer science, Southern Federal University (SFedU). Research interests: software code analysis and investigation. varlamov@sfedu.ru; 8a, Milchakov St., 344090, Rostov-on-Don, Russia; office phone: +7(863)297-5111.

Kosolapov Yury — Ph.D., Associate professor of the department, Department of algebra and discrete mathematics, i.i. vorovich institute of mathematics, mechanics, and computer science, Southern Federal University (SFedU). Research interests: error-correcting codes in cryptography and steganography, theoretical and practical code obfuscation. The number of publications — 110. yvkosolapov@sfedu.ru; 8a, Milchakov St., 344090, Rostov-on-Don, Russia; office phone: +7(863)297-5111.

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 опускается помещать рисунки, таблицы, листинги и формулы. Правила их оформления подробно рассмотрены на нашем сайте в разделе «Руководство для авторов».

Author guidelines

Interaction between each potential author and the Editorial board is realized through the personal account on the website of the journal "Informatics and Automation" <http://ia.spcras.ru/>. At the registration the authors are requested to fill out all data fields in the proposed form. The submissions should be prepared using MS Word 2007, LaTeX. The text of the paper in the main part should not exceed 30 pages. Pages are not numbered; hyphenations are not allowed. Certain figures, tables, listings and formulas are allowed in the main section, and their typography is considered in more detail at the journal web.

Signed to print 06.10.2025. Passed for print 07.10.2025.

Printed in Publishing center GUAP.

Address: 67 litera A, B. Morskaya, St. Petersburg, 190000, Russia

Founder and Publisher: SPC RAS.

Address: 39 litera A, 14th Line V.O., St. Peterburg, 199178, Russia.

The journal is registered in the Federal Service for Supervision of Communications,
Information Technology, and Mass Media,

Registration Certificate (registration number) ПИ № ФС77-79228 dated September 25, 2020

Subscription Index П5513, Russian Post Catalog

Подписано к печати 06.10.2025. Дата выхода в свет 07.10.2025.

Формат 60х90 1/16. Усл. печ. л. 16,3. Заказ № 326. Тираж 300 экз., цена свободная.

Отпечатано в Редакционно-издательском центре ГУАП.

Адрес типографии: Б. Морская, д. 67, лит. А, г. Санкт-Петербург, 190000, Россия

Учредитель и издатель: СПб ФИЦ РАН.

Адрес учредителя и издателя: 14-я линия В.О., д. 39, лит. А, г. Санкт-Петербург, 199178, Россия

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.

Подписной индекс П5513 по каталогу «Почта России»