DOI 10.15622/ia.25.2.6

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

INTELLIGENT FAULT DETECTION AND ISOLATION BASED ON NARX NEURAL NETWORKS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

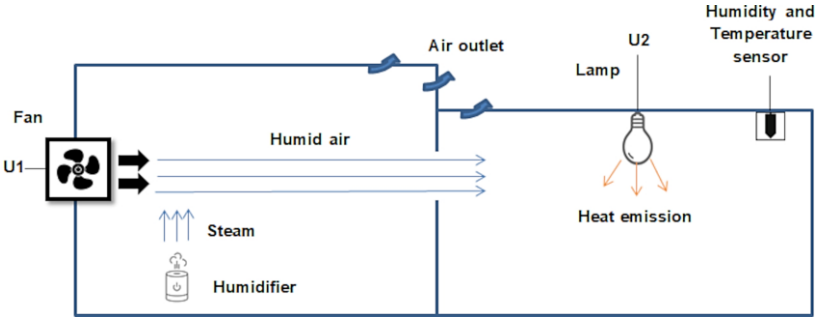

Fig. 1. Schematic representation of the incubator design

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

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

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

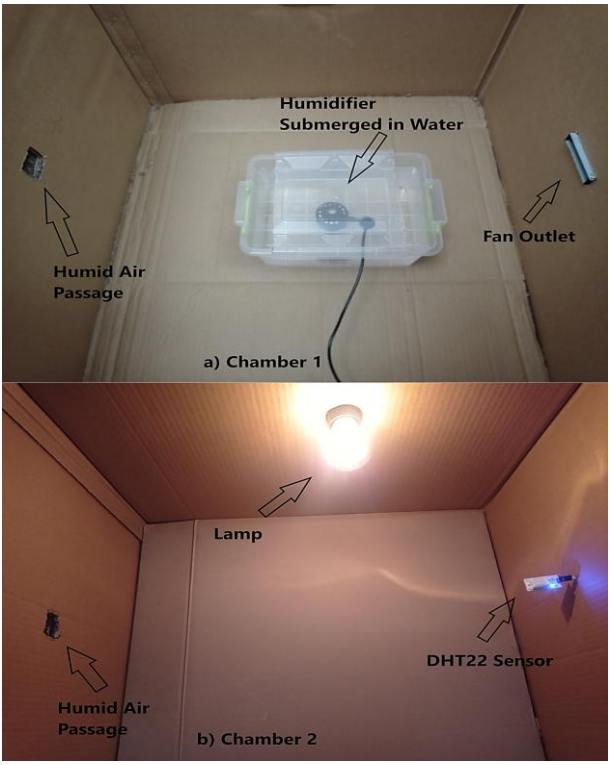

Fig. 2. Real images of the custom incubator: a) chamber 1, b) chamber 2

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

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

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

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

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

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

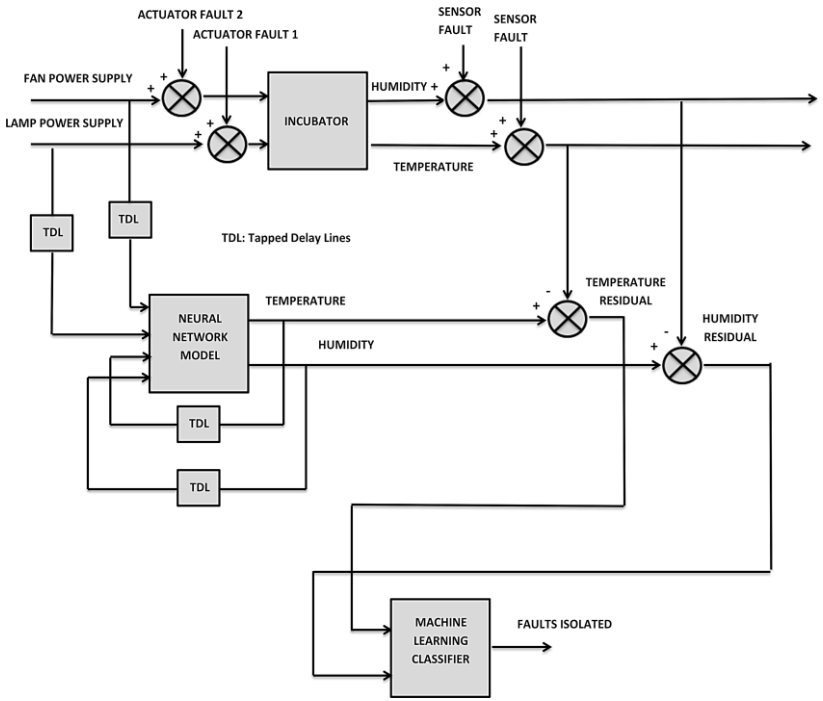

Fig. 3. Schematic representation of the FDI methodology

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

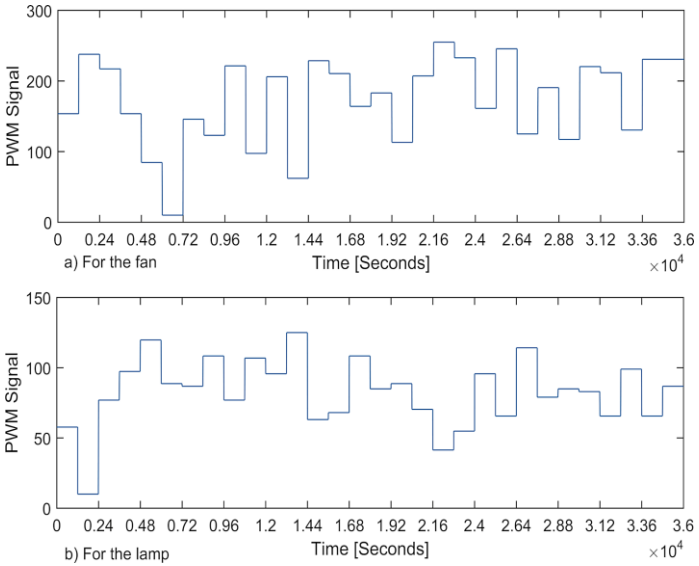

Fig. 4. Excitation signals: a) for the fan, b) for the lamp

The data were divided as follows:

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

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

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

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

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

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

For the MLP network, the following configuration was used:

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

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

– **Number of epochs:** 1000 epochs.

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

For the Elman network, the following configuration was used:

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

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

– **Number of epochs:** 201 epochs.

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

For the LSTM network, the following configuration was used:

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

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

– **Number of epochs:** 247 epochs.

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

For the GRU network, the following configuration was used:

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

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

– **Number of epochs:** 139 epochs.

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

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

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

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

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

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

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

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

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

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

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

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

The configuration of our neural classifier was determined as follows:

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

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

– **Number of epochs:** 1000 epochs.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

– **Root Mean Squared Error:** This metric computes the square root of the average of squared differences between predicted and actual values. It is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \quad (12)$$

RMSE penalizes larger errors more heavily, making it sensitive to significant deviations.

– **Mean Absolute Percentage Error:** This metric expresses the prediction error in relative terms, regardless of the scale of the data. It is calculated using the following formula:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \quad (13)$$

Table 1 presents the performance metrics of the models for the humidity and temperature outputs. A notable difference can be observed between the linear ARX model and the NARX neural networks. The ARX model exhibited higher error values for both outputs, with MAE values of 0.87 and 0.28 for humidity and temperature, respectively. This result indicates the limitation of linear modeling approaches, such as ARX, in capturing the behavior of systems with nonlinear dynamics.

Table 1. Model performance comparison

Model	Output	MAE	RMSE	MAPE
ARX	Humidity	0.87	1.04	1.25
	Temperature	0.28	0.34	0.85
MLP-NARX	Humidity	0.15	0.21	0.21
	Temperature	0.08	0.12	0.24
ELMAN-NARX	Humidity	0.29	0.37	0.42
	Temperature	0.09	0.12	0.29
LSTM-NARX	Humidity	0.24	0.30	0.33
	Temperature	0.10	0.13	0.32
GRU-NARX	Humidity	0.30	0.38	0.44
	Temperature	0.09	0.12	0.29

In contrast, the NARX models achieved low error metrics, demonstrating their effectiveness in modeling both outputs. Furthermore, the small differences between the RMSE and MAE values for these models indicate the absence of significant error peaks, as RMSE penalizes large errors more heavily. Performance differences among the NARX neural networks are more pronounced for humidity prediction, while temperature predictions remain relatively similar. The comparative study showed that the MLP-NARX model achieved the best results, with MAE values of 0.15 and 0.08 for humidity and temperature, respectively. The LSTM-NARX model ranked second, with MAE values of 0.24 for humidity and 0.10 for temperature. The Elman-NARX and GRU-NARX models demonstrated very similar, yet slightly lower, performance, with respective MAE values of 0.29 and 0.30 for humidity and an MAE of 0.09 for temperature. The strong performance of these different implementations of the NARX model highlights its suitability for modeling nonlinear dynamics.

Figure 5 illustrates the modeling results obtained with the selected MLP-NARX model by comparing the actual and predicted outputs of the incubator. The figure shows the last 1200 samples of the training dataset and 1680 samples from the test set, with actual and predicted values plotted for each output. The close alignment between the curves visualizes the low prediction error.

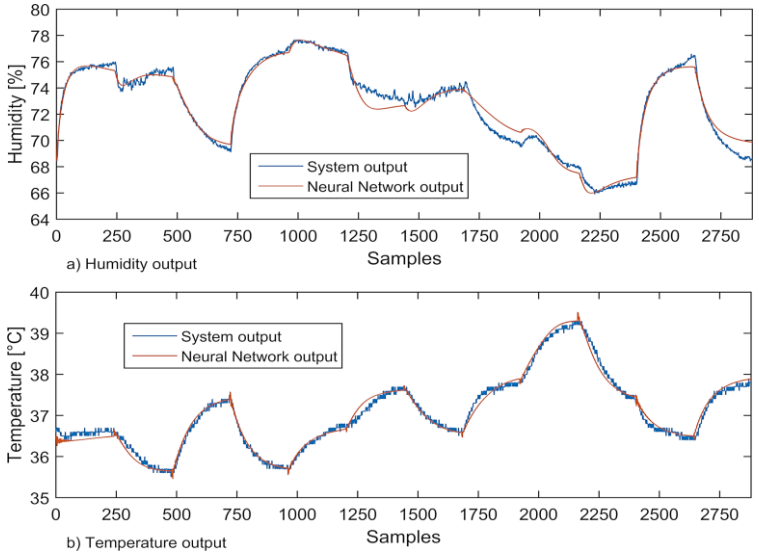

Fig. 5. MLP-NARX Modeling results: comparison between actual and predicted outputs: a) Humidity output, b) Temperature output

7.2. Classification results. The classifiers were evaluated using a structured dataset comprising a total of 2160 samples. The first 250 samples were from the fault-free condition, followed by 250 samples representing the sensor fault condition. An additional 220 healthy-state samples were then included. Subsequently, the dataset included 720 samples for actuator fault 2 and 720 samples for actuator fault 1. Table 2 lists the data samples with their corresponding system states.

Table 2. Data samples with their corresponding system states

Data samples	System state
1-250	Healthy system
251-500	Sensor fault
501-720	Healthy system
721-1440	Actuator fault 2
1441-2160	Actuator fault 1

The fault isolation was achieved using five machine learning algorithms: MLP, SVM, KNN, LDA, and XGBoost. Their effectiveness for the fault classification task was evaluated through a comparative study based on the following performance:

– **Accuracy:** It measures the overall proportion of correct predictions, providing a global indicator of the classifier's performance. It is calculated using the following formula:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}. \quad (14)$$

– **Precision:** It measures the proportion of true positive predictions cases among all positive predictions. It is calculated using the following formula:

$$Precision = \frac{TP}{TP + FP}, \quad (15)$$

where, for a given class, True Positives (TP) denote samples of the class correctly identified; True Negatives (TN) are samples from other classes correctly rejected; False Positives (FP) correspond to samples incorrectly assigned to the class; and False Negatives (FN) are samples of the class that the model failed to recognize. High precision indicates a low rate of false positive errors.

– **Recall:** It measures the proportion of actual positive cases that are correctly identified. It is calculated using the following formula:

$$Recall = \frac{TP}{TP + FN}. \quad (16)$$

High recall indicates that the model successfully captures most of the relevant positive cases.

– **F_1-score:** It is the harmonic mean of precision and recall. It is calculated using the following formula:

$$F_1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (17)$$

It provides a balanced measure between precision and recall, making it particularly useful when both false positives and false negatives are critical, as is the case in our FDI task.

Precision, recall, and F_1-score were calculated for each class and then averaged using the macro-averaging method, which assigns equal weight to all classes.

Table 3 presents the performance metrics of the different ML algorithms for the fault classification task. The comparative study of the different classifiers showed that the MLP classifier achieved the highest results across all metrics, with an accuracy of 95.8%, precision of 96.5%, recall of 96.8%, and F_1-score of 96.6%. These results indicate that the MLP is both highly accurate in its predictions and balanced in terms of precision and recall, making it effective at minimizing both false positives and false negatives, which is crucial for our FDI task.

Table 3. Performance comparison of machine learning classifiers

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)
MLP	95.8	96.5	96.8	96.6
LDA	88.8	86.8	89.4	86.1
XGBOOST	95.4	96.1	96.5	96.2
SVM	90.8	93.4	93.1	92.7
KNN	86.9	91.7	90.2	89.6

The XGBoost classifier was the second-best performer, with an accuracy of 95.4% and an F_1-score of 96.2%. It demonstrated competitive

performance, confirming its suitability for classification tasks, particularly in handling nonlinear relationships.

The SVM also achieved strong results, with an F_1-score of 92.7%. However, it underperformed relative to the top two methods, indicating a lower ability to capture complex data patterns.

LDA and KNN achieved comparatively lower performance. LDA assumes linear class boundaries, which limits its effectiveness on complex nonlinear data, while KNN struggles to capture complex and global relationships within the dataset. They exhibited contrasting performance patterns. LDA tended to favor the majority class, resulting in higher accuracy but lower precision, recall, and F_1-score. The low F_1-score for LDA is a direct consequence of the imbalance between its precision and recall. Conversely, KNN produced a more balanced distribution of predictions across classes, which boosted its precision, recall, and F_1-score but resulted in a drop in overall accuracy.

In conclusion, the results demonstrate that the MLP classifier, owing to its strong ability to capture complex nonlinear relationships, achieved the best fault classification performance, underscoring its high potential for practical FDI applications.

8. Conclusion. In this paper, an intelligent FDI methodology was proposed, using an incubator system as a case study. The design and implementation of the incubator were presented. The complex nonlinear system dynamics were modeled using a NARX neural network in a parallel structure. A comparative study was conducted among different neural networks implementing this model—MLP, LSTM, GRU, and Elman—to identify the best-performing architecture. A linear ARX model was also estimated for comparison with these NARX neural networks to validate their use within the proposed approach. The obtained residuals were then used for FDI using five classification methods: MLP, XGBoost, SVM, KNN, and LDA, whose performance was compared. The results demonstrated the effectiveness of the proposed FDI methodology. The NARX models exhibited strong modeling performance, outperforming the ARX model, with the MLP-NARX attaining the best results. The comparative study of classifiers showed that the MLP classifier achieved the best performance across all evaluation metrics, further highlighting its potential for FDI system.

References

1. Mujcic E., Drakulic U. Design and Implementation of Fuzzy Control System for Egg Incubator based on IoT Technology. IOP Conference Series: Materials Science and Engineering. 2021. vol. 1208. no. 1. DOI: 10.1088/1757-899X/1208/1/012038.

2. Mitchaothai J., Lertpatarakomol R., Trairatapiwan T., Lukkananukool A. Influence of Incubation Temperature and Relative Humidity on the Egg Hatchability Pattern of Two-Spotted (Gryllus bimaculatus) and House (Acheta domesticus) Crickets. Animals. 2024. vol. 14. no. 15. DOI: 10.3390/ani14152176.
3. Noiva R.M., Menezes A.C., Peleteiro M.C. Influence of Temperature and Humidity Manipulation on Chicken Embryonic Development. BMC Veterinary Research. 2014. vol. 10. pp. 1–10. DOI: 10.1186/s12917-014-0234-3.
4. Yakimov V.L., Maltsev G.N. Hybrid Network Structures and Their Use in Diagnosing Complex Technical Systems. Informatics and Automation. 2022. vol. 21. no. 1. pp. 126–160. DOI: 10.15622/ia.2022.21.5.
5. Isermann R. Model-Based Fault-Detection and Diagnosis – Status and Applications. Annual Reviews in Control. 2005. vol. 29. no. 1. pp. 71–85. DOI: 10.1016/j.arcontrol.2004.12.002.
6. Patton R.J. Fault Detection and Diagnosis in Aerospace Systems Using Analytical Redundancy. Computer Control Engineering Journal. 1991. vol. 2. no. 3. pp. 127–136. DOI: 10.1049/cce:19910031.
7. Bernardi E., Adam E.J. Observer-Based Fault Detection and Diagnosis Strategy for Industrial Processes. Journal of the Franklin Institute. 2020. vol. 357. no. 14. pp. 10054–10081. DOI: 10.1016/j.jfranklin.2020.07.046.
8. Mo J., Qin D., Liu Y. Unknown Input Observer-Based Fault Diagnosis of Speed Sensors in Dual Clutch Transmission. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering. 2022. vol. 237. no. 7. pp. 1710–1720. DOI: 10.1177/09544070221095286.
9. Khentout N., Salhi H., Magrotti G., Merrouche D. Fault Monitoring and Accommodation of the Heat Exchanger Parameters of Triga-Mark II Nuclear Research Reactor Using Model-Based Analytical Redundancy. Progress in Nuclear Energy. 2018. vol. 109. pp. 97–112. DOI: 10.1016/j.pnucene.2018.02.019.
10. Jihani N., Kabbaj M.N., Benbrahim M. Kalman Filter Based Sensor Fault Detection in Wireless Sensor Network for Smart Irrigation. Results in Engineering. 2023. vol. 20. DOI: 10.1016/j.rineng.2023.101395.
11. Patan K. Artificial Neural Networks for the Modeling and Fault Diagnosis of Technical Processes. Springer, 2008. 206 p.. DOI: 10.1007/978-3-540-79872-9.
12. Patan K., Korbicz J. Artificial Neural Networks in Fault Diagnosis. Berlin, Heidelberg: Springer Berlin Heidelberg. 2004. pp. 333–379. DOI: 10.1007/978-3-642-18615-8_9.
13. Yu D.L., Hamad A., Gomm J.B., Sangha M.S. Dynamic Fault Detection and Isolation for Automotive Engine Air Path by Independent Neural Network Model. International Journal of Engine Research. 2014. vol. 15. no. 1. pp. 87–100. DOI: 10.1177/1468087412461267.
14. Mohd Amiruddin A.A., Zabiri H., Taqvi S.A.A., Tufa L.D. Neural Network Applications in Fault Diagnosis and Detection: An Overview of Implementations in Engineering-Related Systems. Neural Computing and Applications. 2020. vol. 32. no. 2. pp. 447–472. DOI: 10.1007/s00521-018-3911-5.
15. Atmane M.A., Boudebouda O., Zatla H., Nouar S.F., Tolbi B., Djeriri Y., Bouhamama M. Fault Diagnosis: An Integrated Methodology Based on Neural Networks. Proceedings of 3rd International Conference on Advanced Electrical Engineering (ICAEE'2024). IEEE, 2024. pp. 1–6. DOI: 10.1109/ICAEE61760.2024.10783413.
16. Nasser R., Azar A.T., Humaidi A.J., Al-Mhdawi A.K., Ibraheem I.K. Intelligent Fault Detection and Identification Approach for Analog Electronic Circuits Based on Fuzzy Logic Classifier. Electronics. 2021. vol. 10. no. 23. DOI: 10.3390/electronics10232888.

17. Yang L., Gao L., Luo X., Hao Y., Zhang Z., Jin Y., Zhang J. Improved Method for Fault Diagnosis of Oil-Immersed Transformers Based on Simulation Test Platform. IEEE Transactions on Dielectrics and Electrical Insulation. 2025. vol. 32. no. 1. pp. 571–580. DOI: 10.1109/TDEI.2024.3418388.
18. Venkata P., Pandya V., Vala K., Sant A.V. Support Vector Machine for Fast Fault Detection and Classification in Modern Power Systems Using Quarter Cycle Data. Energy Reports. 2022. vol. 8. pp. 92–98. DOI: 10.1016/j.egyr.2022.10.279.
19. Tuerxun W., Chang X., Hongyu G., Zhijie J., Huajian Z. Fault Diagnosis of Wind Turbines Based on a Support Vector Machine Optimized by the Sparrow Search Algorithm. IEEE Access. 2021. vol. 9. pp. 69307–69315. DOI: 10.1109/ACCESS.2021.3075547.
20. Cho S., Choi M., Gao Z., Moan T. Fault Detection and Diagnosis of a Blade Pitch System in a Floating Wind Turbine Based on Kalman Filters and Artificial Neural Networks. Renewable Energy. 2021. vol. 169. pp. 1–13. DOI: 10.1016/j.renene.2020.12.116.
21. Wang Y., Wen C., Wu X. Fault Detection and Isolation of Floating Wind Turbine Pitch System Based on Kalman Filter and Multi-Attention 1DCNN. Systems Science & Control Engineering. 2024. vol. 12. no. 1. DOI: 10.1080/21642583.2024.2362169.
22. Yu D.L., Gomm J.B., Williams D. Sensor Fault Diagnosis in a Chemical Process via RBF Neural Networks. Control Engineering Practice. 1999. vol. 7. no. 1. pp. 49–55. DOI: 10.1016/S0967-0661(98)00167-1.
23. Liu X., He H. Fault Diagnosis for TE Process Using RBF Neural Network. IEEE Access. 2021. vol. 9. pp. 118453–118460. DOI: 10.1109/ACCESS.2021.3107360.
24. Rahimilarki R., Gao Z., Jin N., Zhang A. Convolutional Neural Network Fault Classification Based on Time-Series Analysis for Benchmark Wind Turbine Machine. Renewable Energy. 2022. vol. 185. pp. 916–931. DOI: 10.1016/j.renene.2021.12.056.
25. Choudhary A., Mian T., Fatima S. Convolutional Neural Network Based Bearing Fault Diagnosis of Rotating Machine Using Thermal Images. Measurement. 2021. vol. 176. DOI: 10.1016/j.measurement.2021.109196.
26. Zhang Y., Zhou T., Huang X., Cao L., Zhou Q. Fault Diagnosis of Rotating Machinery Based on Recurrent Neural Networks. Measurement. 2021. vol. 171. DOI: 10.1016/j.measurement.2020.108774.
27. Zhu J., Jiang Q., Shen Y., Qian C., Xu F., Zhu Q. Application of Recurrent Neural Network to Mechanical Fault Diagnosis: A Review. Journal of Mechanical Science and Technology. 2022. vol. 36. no. 2. pp. 527–542. DOI: 10.1007/s12206-022-0102-1.
28. An Y., Sun X., Ren B., Li H., Zhang M. A Data-driven Method for IGBT Open-circuit Fault Diagnosis for the Modular Multilevel Converter based on a modified Elman Neural Network. Energy Reports. 2022. vol. 8. pp. 80–88. DOI: 10.1016/j.egyr.2022.08.024.
29. Nguyen V.-Q., Vu M.-H., Pham V.-T., Tran T.-T. A Deep Learning Approach Based on MLP-Mixer Models for Bearing Fault Diagnosis. Proceedings of International Conference on System Science and Engineering (ICSSE'2023). IEEE, 2023. pp. 16–21. DOI: 10.1109/ICSSE58758.2023.10227218.
30. Khoualdia T., Lakehal A., Chelli Z., Khoualdia K., Nessaib K. Optimized Multi Layer Perceptron Artificial Neural Network Based Fault Diagnosis of Induction Motor Using Vibration Signals. Diagnostyka. 2021. vol. 22. no. 1. pp. 65–74. DOI: 10.29354/diag/133091.
31. Naimi A., Deng J., Shimjith S.R., Arul A.J. Fault Detection and Isolation of a Pressurized Water Reactor based on Neural Network and K-Nearest Neighbor. IEEE Access. 2022. vol. 10. pp. 17113–17121. DOI: 10.1109/ACCESS.2022.3149772.

32. Ma Y., Liu H., Zhu Y., Wang F., Luo Z. The NARX Model-Based System Identification on Nonlinear, Rotor-Bearing Systems. Applied Sciences. 2017. vol. 7. no. 9. pp. 911. DOI: 10.3390/app7090911.
33. Choe H.O., Lee M.H. Artificial Intelligence-based Fault Diagnosis and Prediction for Smart Farm Information and Communication Technology Equipment. Agriculture. 2023. vol. 13. no. 11. DOI: 10.3390/agriculture13112124.
34. Taqvi S.A., Tufa L.D., Zabiri H., Maulud A.S., Uddin F. Fault Detection in Distillation Column Using NARX Neural Network. Neural Computing and Applications. 2020. vol. 32. no. 8. pp. 3503–3519. DOI: 10.1007/s00521-018-3658-z.
35. Mustafaraj G., Lowry G., Chen J. Prediction of Room Temperature and Relative Humidity by Autoregressive Linear and Nonlinear Neural Network Models for an Open Office. Energy and Buildings. 2011. vol. 43. no. 6. pp. 1452–1460. DOI: 10.1016/j.enbuild.2011.02.007.
36. Sani M.G., Abdul Wahab N., Sam Y.M., Samsudin S.I., Jamaludin I.W. Comparison of NARX Neural Network and Classical Modelling Approaches. Applied mechanics and Materials. 2014. vol. 554. pp. 360–365. DOI: 10.4028/www.scientific.net/AMM.554.360.
37. Lin Y.P. ARX and NARX Modeling and Control of a Continuous UV/H2O2 Photoreactor for the Aqueous PVA Degradation. Master's Thesis. Toronto Metropolitan University, 2024. 171 p. DOI: 10.32920/25761546.
38. Ling T.G., Rahmat M.F., Husain A.R. A Comparative Study of Linear ARX and Nonlinear ANFIS Modeling of an Electro-hydraulic Actuator System. Jurnal Teknologi (Sciences & Engineering). 2014. vol. 67. no. 5. pp. 1–8. DOI: 10.11113/jt.v67.2833.
39. Renteria-Mena J.B., Plaza D., Giraldo E. Multivariable NARX based Neural Networks Models for Short-term Water Level Forecasting. Engineering Proceedings. 2023. vol. 39. no. 1. DOI: 10.3390/engproc2023039060.
40. Delcroix B., Ny J.L., Bernier M., Azam M., Qu B., Venne J.S. Autoregressive Neural Networks with Exogenous Variables for Indoor Temperature Prediction in Buildings. Building simulation. 2021. vol. 14. no. 1. pp. 165–178. DOI: 10.1007/s12273-019-0597-2.
41. Ramirez C., Acuna G. Forecasting Cash Demand in ATM using Neural Networks and Least Square Support Vector Machine. Proceedings of 16th Iberoamerican Congress on Pattern Recognition (CIARP). Springer Berlin Heidelberg, 2011. pp. 515–522. DOI: 10.1007/978-3-642-25085-9_61.
42. Shao Y., Zhao L. NARX-GA-Elman Method for Mach Number Prediction of Wind Tunnel Flow Field. Instrumentation. 2023. vol. 10. no. 4. DOI: 10.15878/j.cnki.instrumentation.2023.04.004.
43. Nikentari N., Wei H.L. Tide Level Prediction Using NARX-based Recurrent Neural Networks. Proceeding of 27th International Conference on Automation and Computing (ICAC). IEEE, 2022. pp. 1–6. DOI: 10.1109/ICAC55051.2022.9911163.
44. Freitas V.C.G.D., Araujo V.G.D., Crisostomo D.C.D.C., Lima G.F.D., Neto A.D.D., Salazar A.O. Velocity Prediction of a Pipeline Inspection Gauge (PIG) with Machine Learning. Sensors. 2022. vol. 22. no. 23. DOI: 10.3390/s22239162.
45. Ghate V.N., Dudul S.V. Optimal MLP Neural Network Classifier for Fault Detection of Three Phase Induction Motor. Expert Systems with Applications. 2010. vol. 37. no. 4. pp. 3468–3481. DOI: 10.1016/j.eswa.2009.10.041.
46. Teler K., Skowron M., Orłowska-Kowalska T. Implementation of MLP-Based Classifier of Current Sensor Faults in Vector-Controlled Induction Motor Drive. IEEE Transactions on Industrial Informatics. 2024. vol. 20. no. 4. pp. 5702–5713. DOI: 10.1109/TII.2023.3336348.

47. Lin W., Yang C., Lin J., Tsay M.A. Fault Classification Method by RBF Neural Network with OLS Learning Procedure. IEEE Power Engineering Review. 2001. vol. 21. no. 8. DOI: 10.1109/MPER.2001.4311561.
48. Yang P., Wang T., Yang H., Meng C., Zhang H., Cheng L. The Performance of Electronic Current Transformer Fault Diagnosis Model: Using an Improved Whale Optimization Algorithm and RBF Neural Network. Electronics. 2023. vol. 12. no. 4. DOI: 10.3390/electronics12041066.
49. Jeong K., Choi S.B., Choi H. Sensor Fault Detection and Isolation Using a Support Vector Machine for Vehicle Suspension Systems. IEEE Transactions on Vehicular Technology. 2020. vol. 69. no. 4. pp. 3852–3863. DOI: 10.1109/TVT.2020.2977353.
50. Ramdani O., Beddek K., Haddouche R., Zerrougui M., Chouider N. SVM-Based Approach Fault Detection for PMSG-Wind Energy Conversion System. Journal of Engineering Research. 2025. vol. 13. no. 4. pp. 3378–3393. DOI: 10.1016/j.jer.2025.01.001.
51. Zhou N., Shao Q., Zhou J., Changjie H. Fault Classification of Proton Exchange Membrane Fuel Cells for Vehicles based on XGBoost. Proceedings of 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2021. pp. 1054–1058. DOI: 10.1109/ICBAIE52039.2021.9389943.
52. Hasan S., Toufikuzzaman M. Fault Occurrence Detection and Classification of Fault Type in Electrical Power Transmission Line with Machine Learning Algorithms. International Journal on Electrical Engineering and Informatics. 2022. vol. 14. no. 3. DOI: 10.15676/ijeei.2022.14.3.9.
53. Mehta A., Goyal D., Choudhary A., Pabla B.S., Belghith S. Machine Learning-Based Fault Diagnosis of Self-Aligning Bearings for Rotating Machinery Using Infrared Thermography. Mathematical Problems in Engineering. 2021. vol. 2021. no. 1. DOI: 10.1155/2021/9947300.
54. Shaikh F.A., Kamboh M.Z., Alvi B.A., Khan S., Khan F.M. Condition-Based Health Monitoring of Electrical Machines Using DWT and LDA Classifier. Sir Syed University Research Journal of Engineering and Technology. 2022. vol. 12. no. 2. pp. 95–100. DOI: 10.33317/ssurj.513.
55. Altayef E., Anayi F., Packianather M., Benmahamed Y., Kherif O. Detection and Classification of Lamination Faults in A 15 kVA Three-Phase Transformer Core using SVM, KNN and DT Algorithms. IEEE Access. 2022. vol. 10. pp. 50925–50932. DOI: 10.1109/ACCESS.2022.3174359.
56. Shinde P.V., Desavale R.G., Jadhav P.M., Sawant S.H. A Multi Fault Classification in a Rotor-Bearing System using Machine Learning Approach. Journal of the Brazilian Society of Mechanical Sciences and Engineering. 2023. vol. 45. no. 2. DOI: 10.1007/s40430-023-04015-1.
57. Rakhmawati R., Murdianto F.D., Luthfi A., Rahman A.Y. Thermal Optimization on Incubator using Fuzzy Inference System based IoT. Proceedings of International Conference of Artificial Intelligence and Information Technology (ICAIIT). IEEE, 2019. pp. 464–468. DOI: 10.1109/ICAIIT.2019.8834530.
58. Febriani A., Wahyuni R., Mardeni M., Irawan Y., Melyanti R. Improved Hybrid Machine and Deep Learning Model for Optimization of Smart Egg Incubator. Journal of Applied Data Sciences. 2024. vol. 5. no. 3. pp. 1052–1068. DOI: 10.47738/jads.v5i3.304.
59. Easwaran A., Arvindan P., Dhanyasree E., Surya R., Selvakumar S. Internet of Things Enabled Smart Animal Farm Prototype. Journal of Physics: Conference Series. 2021. vol. 2070. no. 1. DOI: 10.1088/1742-6596/2070/1/012115.
60. Guzman-Zabala J.V., Castro-Martin A.P. Smart and Semi-industrial Egg Incubator with Remote Monitoring Using LoRa Technology. Proceedings of International Conference on

Computer Science, Electronics and Industrial Engineering (CSEI). Springer Nature Switzerland, 2024. pp. 522–540. DOI: 10.1007/978-3-031-70981-4_35.
61. Roshanghiyasi H., Ahmad A.H., Hosseinpour S., Jafari A., Mousazadeh H., Asadollahzadeh A. Monitoring and Controlling the Chicken Incubation Process Using the Internet of Things System. Journal of Agricultural Mechanization. 2022. vol. 7. no. 1. pp. 73–77. DOI: 10.22034/jam.2022.15710.
62. Radhakrishnan K., Jose N., Sanjay S.G., Cherian T., Vishnu K.R. Design and Implementation of a Fully Automated Egg Incubator. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2014. vol. 3. no. 2. pp. 7686–7690.
63. Uzoigwe L.O., Ekezie J.C. Egg Incubator Control System with Short Message Service (sms) Fault Analysis Alert. Journal of Agriculture and Food Sciences. 2013. vol. 11. no. 2. pp. 45–68. DOI: 10.4314/jafs.v11i2.5.
64. Stull R. Practical Meteorology: An Algebra-based Survey of Atmospheric Science. University of British Columbia, 2017. 940 p.
65. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Representations by Back-Propagating Errors. Nature. 1986. vol. 323. no. 6088. pp. 533–536. DOI: 10.1038/323533a0.
66. Elman J.L. Finding Structure in Time. Cognitive science. 1990. vol. 14. no. 2. pp. 179–211. DOI: 10.1207/s15516709cog1402_1.
67. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural computation. 1997. vol. 9. no. 8. pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
68. Cho K., Van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014. pp. 1724–1734. DOI: 10.3115/v1/D14-1179.
69. MacKay D.J.C. Bayesian Interpolation. Neural Computation. 1992. vol. 4. no. 3. pp. 415–447. DOI: 10.1162/neco.1992.4.3.415.
70. Londhe A.S., Ingale A.S., Khadse C.B. Bayesian Regularization Neural Network-based Fault Detection System in HVDC Transmission System. Proceedings of International Conference on Smart Technologies for Energy, Environment, and Sustainable Development (ICSTEESD). Springer Nature, 2022. pp. 601–607. DOI: 10.1007/978-981-16-6875-3_48.
71. Hassan M.S., Kamal K., Ratlamwala T.A.H. Fault Classification of Power Plants using Artificial Neural Network. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. 2022. vol. 44. no. 3. pp. 7665–7680. DOI: 10.1080/15567036.2022.2113936.
72. Liu D.C., Nocedal J. On the Limited Memory BFGS Method for Large Scale Optimization. Mathematical Programming. 1989. vol. 45. no. 1. pp. 503–528. DOI: 10.1007/BF01589116
73. Ljung L. System Identification: Theory for the User. Prentice Hall, 1999. pp. 640.

Atmane Mohamed — Ph.D. student, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: fault detection and isolation, artificial intelligence, automation. The number of publications — 1. mohamed.atmane@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Zatla Hicham — Ph.D., Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: automatic control, artificial intelligence, human-machine systems. The number of publications — 8.

hicham.zatla@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Tolbi Bilal — Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: artificial intelligence, hybrid systems, systems control. The number of publications — 15. bilal.tolbi@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Nouar Souad — Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: physical system modeling, numerical simulation. The number of publications — 2. souadfadila.nouar@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Bouhamama Mohamed — Senior lecturer, Department of automatic, faculty of electrical engineering, Djilali Liabes University of Sidi Bel Abbes. Research interests: electrical engineering, environmental technologies. The number of publications — 14. mohammed.bouhamama@univ-sba.dz; PB 89, 22000, Sidi Bel Abbes, Algeria; office phone: +213(4870)01-38.

Acknowledgements. This research is supported by Directorate General for Scientific Research and Technological Development (DGRSDT).

DOI 10.15622/ia.25.2.6

М.А. АТМАНЕ, Х. ЗАТЛА, Б. ТОЛБИ, С.Ф. НУАР, М. БУХАМАМА

ИНТЕЛЛЕКТУАЛЬНОЕ ОБНАРУЖЕНИЕ И ИЗОЛЯЦИЯ НЕИСПРАВНОСТЕЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ NARX

Атмане М.А., Затла Х., Толби Б., Нуар С.Ф., Бухамама М. **Интеллектуальное обнаружение и изоляция неисправностей на основе нейронных сетей NARX.**

Аннотация. Интеллектуальные системы стали неотъемлемым компонентом современных технологических ландшафтов. Вопрос их надежности является крайне важным, поскольку возникновение неисправностей способно катастрофически сказаться на функционировании и итоговой эффективности системы, вследствие чего обнаружение и изоляция неисправностей (FDI) становится критически значимой задачей. Реализация этой задачи осложняется присущей таким системам сложной нелинейной динамикой. В данной работе в рамках решения указанной проблемы предлагается методология интеллектуального обнаружения и изоляции неисправностей; в качестве репрезентативного примера рассматривается система инкубатора. Предлагаемый метод использует нелинейную авторегрессионную экзогенную (NARX) нейронную сеть в параллельной структуре для моделирования сложной нелинейной динамики системы. Были сравнены различные реализации модели NARX, основанные на многослойном перцептроне (MLP), сети долгой краткосрочной памяти (LSTM), вентильном рекуррентном блоке (GRU) и сети Элмана, для оценки их эффективности моделирования и определения лучшей модели. Полученные расхождения между прогнозами модели и фактическими значениями называются остаточными величинами. Для классификации неисправностей было проведено сравнительное исследование пяти методов машинного обучения (ML): многослойного перцептрона, экстремального градиентного бустинга (XGBoost), метода опорных векторов (SVM), метода k-ближайших соседей (KNN) и линейного дискриминантного анализа (LDA). Данные методы анализируют остаточные величины, чтобы идентифицировать конкретную неисправность из заранее определенного множества возможных, включая неисправности исполнительных механизмов и датчиков. Полученные результаты демонстрируют эффективность интеллектуальной методологии FDI, представленной в данной работе. Модели NARX продемонстрировали высокую эффективность моделирования, причем гибридная модель MLP-NARX показала наилучшие результаты, превзойдя остальные архитектуры. Сравнительный анализ классификаторов выявил различия в эффективности пяти рассматриваемых методов, при этом классификатор на основе многослойного перцептрона достиг наивысших показателей по всем оценочным метрикам. Это свидетельствует о его пригодности для практического применения в задачах диагностики неисправностей, что обусловлено его высокой способностью улавливать сложные нелинейные взаимосвязи в данных.

Ключевые слова: обнаружение и изоляция неисправностей, инкубатор, многослойный перцептрон, моделирование NARX, генерация и оценка остаточных величин, рекуррентные нейронные сети, классификаторы машинного обучения, интеллектуальные системы.

Литература

1. Mujcic E., Drakulic U. Design and Implementation of Fuzzy Control System for Egg Incubator based on IoT Technology. IOP Conference Series: Materials Science and Engineering. 2021. vol. 1208. no. 1. DOI: 10.1088/1757-899X/1208/1/012038.

2. Mitchaothai J., Lertpatarakomol R., Trairatapiwan T., Lukkananukool A. Influence of Incubation Temperature and Relative Humidity on the Egg Hatchability Pattern of Two-Spotted (Gryllus bimaculatus) and House (Acheta domesticus) Crickets. Animals. 2024. vol. 14. no. 15. DOI: 10.3390/ani14152176.
3. Noiva R.M., Menezes A.C., Peleteiro M.C. Influence of Temperature and Humidity Manipulation on Chicken Embryonic Development. BMC Veterinary Research. 2014. vol. 10. pp. 1–10. DOI: 10.1186/s12917-014-0234-3.
4. Yakimov V.L., Maltsev G.N. Hybrid Network Structures and Their Use in Diagnosing Complex Technical Systems. Informatics and Automation. 2022. vol. 21. no. 1. pp. 126–160. DOI: 10.15622/ia.2022.21.5.
5. Isermann R. Model-Based Fault-Detection and Diagnosis – Status and Applications. Annual Reviews in Control. 2005. vol. 29. no. 1. pp. 71–85. DOI: 10.1016/j.arcontrol.2004.12.002.
6. Patton R.J. Fault Detection and Diagnosis in Aerospace Systems Using Analytical Redundancy. Computer Control Engineering Journal. 1991. vol. 2. no. 3. pp. 127–136. DOI: 10.1049/cce:19910031.
7. Bernardi E., Adam E.J. Observer-Based Fault Detection and Diagnosis Strategy for Industrial Processes. Journal of the Franklin Institute. 2020. vol. 357. no. 14. pp. 10054–10081. DOI: 10.1016/j.jfranklin.2020.07.046.
8. Mo J., Qin D., Liu Y. Unknown Input Observer-Based Fault Diagnosis of Speed Sensors in Dual Clutch Transmission. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering. 2022. vol. 237. no. 7. pp. 1710–1720. DOI: 10.1177/09544070221095286.
9. Khentout N., Salhi H., Magrotti G., Merrouche D. Fault Monitoring and Accommodation of the Heat Exchanger Parameters of Triga-Mark II Nuclear Research Reactor Using Model-Based Analytical Redundancy. Progress in Nuclear Energy. 2018. vol. 109. pp. 97–112. DOI: 10.1016/j.pnucene.2018.02.019.
10. Jihani N., Kabbaj M.N., Benbrahim M. Kalman Filter Based Sensor Fault Detection in Wireless Sensor Network for Smart Irrigation. Results in Engineering. 2023. vol. 20. DOI: 10.1016/j.rineng.2023.101395.
11. Patan K. Artificial Neural Networks for the Modeling and Fault Diagnosis of Technical Processes. Springer, 2008. 206 p.. DOI: 10.1007/978-3-540-79872-9.
12. Patan K., Korbicz J. Artificial Neural Networks in Fault Diagnosis. Berlin, Heidelberg: Springer Berlin Heidelberg. 2004. pp. 333–379. DOI: 10.1007/978-3-642-18615-8_9.
13. Yu D.L., Hamad A., Gomm J.B., Sangha M.S. Dynamic Fault Detection and Isolation for Automotive Engine Air Path by Independent Neural Network Model. International Journal of Engine Research. 2014. vol. 15. no. 1. pp. 87–100. DOI: 10.1177/1468087412461267.
14. Mohd Amiruddin A.A., Zabiri H., Taqvi S.A.A., Tufa L.D. Neural Network Applications in Fault Diagnosis and Detection: An Overview of Implementations in Engineering-Related Systems. Neural Computing and Applications. 2020. vol. 32. no. 2. pp. 447–472. DOI: 10.1007/s00521-018-3911-5.
15. Atmane M.A., Boudebouda O., Zatla H., Nouar S.F., Tolbi B., Djeriri Y., Bouhamama M. Fault Diagnosis: An Integrated Methodology Based on Neural Networks. Proceedings of 3rd International Conference on Advanced Electrical Engineering (ICAEE'2024). IEEE, 2024. pp. 1–6. DOI: 10.1109/ICAEE61760.2024.10783413.
16. Nasser R., Azar A.T., Humaidi A.J., Al-Mhdawi A.K., Ibraheem I.K. Intelligent Fault Detection and Identification Approach for Analog Electronic Circuits Based on Fuzzy Logic Classifier. Electronics. 2021. vol. 10. no. 23. DOI: 10.3390/electronics10232888.
17. Yang L., Gao L., Luo X., Hao Y., Zhang Z., Jin Y., Zhang J. Improved Method for Fault Diagnosis of Oil-Immersed Transformers Based on Simulation Test Platform.

IEEE Transactions on Dielectrics and Electrical Insulation. 2025. vol. 32. no. 1. pp. 571–580. DOI: 10.1109/TDEI.2024.3418388.
18. Venkata P., Pandya V., Vala K., Sant A.V. Support Vector Machine for Fast Fault Detection and Classification in Modern Power Systems Using Quarter Cycle Data. Energy Reports. 2022. vol. 8. pp. 92–98. DOI: 10.1016/j.egyr.2022.10.279.
19. Tuerxun W., Chang X., Hongyu G., Zhijie J., Huajian Z. Fault Diagnosis of Wind Turbines Based on a Support Vector Machine Optimized by the Sparrow Search Algorithm. IEEE Access. 2021. vol. 9. pp. 69307–69315. DOI: 10.1109/ACCESS.2021.3075547.
20. Cho S., Choi M., Gao Z., Moan T. Fault Detection and Diagnosis of a Blade Pitch System in a Floating Wind Turbine Based on Kalman Filters and Artificial Neural Networks. Renewable Energy. 2021. vol. 169. pp. 1–13. DOI: 10.1016/j.renene.2020.12.116.
21. Wang Y., Wen C., Wu X. Fault Detection and Isolation of Floating Wind Turbine Pitch System Based on Kalman Filter and Multi-Attention 1DCNN. Systems Science & Control Engineering. 2024. vol. 12. no. 1. DOI: 10.1080/21642583.2024.2362169.
22. Yu D.L., Gomm J.B., Williams D. Sensor Fault Diagnosis in a Chemical Process via RBF Neural Networks. Control Engineering Practice. 1999. vol. 7. no. 1. pp. 49–55. DOI: 10.1016/S0967-0661(98)00167-1.
23. Liu X., He H. Fault Diagnosis for TE Process Using RBF Neural Network. IEEE Access. 2021. vol. 9. pp. 118453–118460. DOI: 10.1109/ACCESS.2021.3107360.
24. Rahimilarki R., Gao Z., Jin N., Zhang A. Convolutional Neural Network Fault Classification Based on Time-Series Analysis for Benchmark Wind Turbine Machine. Renewable Energy. 2022. vol. 185. pp. 916–931. DOI: 10.1016/j.renene.2021.12.056.
25. Choudhary A., Mian T., Fatima S. Convolutional Neural Network Based Bearing Fault Diagnosis of Rotating Machine Using Thermal Images. Measurement. 2021. vol. 176. DOI: 10.1016/j.measurement.2021.109196.
26. Zhang Y., Zhou T., Huang X., Cao L., Zhou Q. Fault Diagnosis of Rotating Machinery Based on Recurrent Neural Networks. Measurement. 2021. vol. 171. DOI: 10.1016/j.measurement.2020.108774.
27. Zhu J., Jiang Q., Shen Y., Qian C., Xu F., Zhu Q. Application of Recurrent Neural Network to Mechanical Fault Diagnosis: A Review. Journal of Mechanical Science and Technology. 2022. vol. 36. no. 2. pp. 527–542. DOI: 10.1007/s12206-022-0102-1.
28. An Y., Sun X., Ren B., Li H., Zhang M. A Data-driven Method for IGBT Open-circuit Fault Diagnosis for the Modular Multilevel Converter based on a modified Elman Neural Network. Energy Reports. 2022. vol. 8. pp. 80–88. DOI: 10.1016/j.egyr.2022.08.024.
29. Nguyen V.-Q., Vu M.-H., Pham V.-T., Tran T.-T. A Deep Learning Approach Based on MLP-Mixer Models for Bearing Fault Diagnosis. Proceedings of International Conference on System Science and Engineering (ICSSE'2023). IEEE, 2023. pp. 16–21. DOI: 10.1109/ICSSE58758.2023.10227218.
30. Khoualdia T., Lakehal A., Chelli Z., Khoualdia K., Nessaib K. Optimized Multi Layer Perceptron Artificial Neural Network Based Fault Diagnosis of Induction Motor Using Vibration Signals. Diagnostyka. 2021. vol. 22. no. 1. pp. 65–74. DOI: 10.29354/diag/133091.
31. Naimi A., Deng J., Shimjith S.R., Arul A.J. Fault Detection and Isolation of a Pressurized Water Reactor based on Neural Network and K-Nearest Neighbor. IEEE Access. 2022. vol. 10. pp. 17113–17121. DOI: 10.1109/ACCESS.2022.3149772.
32. Ma Y., Liu H., Zhu Y., Wang F., Luo Z. The NARX Model-Based System Identification on Nonlinear, Rotor-Bearing Systems. Applied Sciences. 2017. vol. 7. no. 9. pp. 911. DOI: 10.3390/app7090911.

33. Choe H.O., Lee M.H. Artificial Intelligence-based Fault Diagnosis and Prediction for Smart Farm Information and Communication Technology Equipment. Agriculture. 2023. vol. 13. no. 11. DOI: 10.3390/agriculture13112124.
34. Taqvi S.A., Tufa L.D., Zabiri H., Maulud A.S., Uddin F. Fault Detection in Distillation Column Using NARX Neural Network. Neural Computing and Applications. 2020. vol. 32. no. 8. pp. 3503–3519. DOI: 10.1007/s00521-018-3658-z.
35. Mustafaraj G., Lowry G., Chen J. Prediction of Room Temperature and Relative Humidity by Autoregressive Linear and Nonlinear Neural Network Models for an Open Office. Energy and Buildings. 2011. vol. 43. no. 6. pp. 1452–1460. DOI: 10.1016/j.enbuild.2011.02.007.
36. Sani M.G., Abdul Wahab N., Sam Y.M., Samsudin S.I., Jamaludin I.W. Comparison of NARX Neural Network and Classical Modelling Approaches. Applied mechanics and Materials. 2014. vol. 554. pp. 360–365. DOI: 10.4028/www.scientific.net/AMM.554.360.
37. Lin Y.P. ARX and NARX Modeling and Control of a Continuous UV/H2O2 Photoreactor for the Aqueous PVA Degradation. Master's Thesis. Toronto Metropolitan University, 2024. 171 p. DOI: 10.32920/25761546.
38. Ling T.G., Rahmat M.F., Husain A.R. A Comparative Study of Linear ARX and Nonlinear ANFIS Modeling of an Electro-hydraulic Actuator System. Jurnal Teknologi (Sciences & Engineering). 2014. vol. 67. no. 5. pp. 1–8. DOI: 10.11113/jt.v67.2833.
39. Renteria-Mena J.B., Plaza D., Giraldo E. Multivariable NARX based Neural Networks Models for Short-term Water Level Forecasting. Engineering Proceedings. 2023. vol. 39. no. 1. DOI: 10.3390/engproc2023039060.
40. Delcroix B., Ny J.L., Bernier M., Azam M., Qu B., Venne J.S. Autoregressive Neural Networks with Exogenous Variables for Indoor Temperature Prediction in Buildings. Building simulation. 2021. vol. 14. no. 1. pp. 165–178. DOI: 10.1007/s12273-019-0597-2.
41. Ramirez C., Acuna G. Forecasting Cash Demand in ATM using Neural Networks and Least Square Support Vector Machine. Proceedings of 16th Iberoamerican Congress on Pattern Recognition (CIARP). Springer Berlin Heidelberg, 2011. pp. 515–522. DOI: 10.1007/978-3-642-25085-9_61.
42. Shao Y., Zhao L. NARX-GA-Elman Method for Mach Number Prediction of Wind Tunnel Flow Field. Instrumentation. 2023. vol. 10. no. 4. DOI: 10.15878/j.cnki.instrumentation.2023.04.004.
43. Nikentari N., Wei H.L. Tide Level Prediction Using NARX-based Recurrent Neural Networks. Proceeding of 27th International Conference on Automation and Computing (ICAC). IEEE, 2022. pp. 1–6. DOI: 10.1109/ICAC55051.2022.9911163.
44. Freitas V.C.G.D., Araujo V.G.D., Crisostomo D.C.D.C., Lima G.F.D., Neto A.D.D., Salazar A.O. Velocity Prediction of a Pipeline Inspection Gauge (PIG) with Machine Learning. Sensors. 2022. vol. 22. no. 23. DOI: 10.3390/s22239162.
45. Ghate V.N., Dudul S.V. Optimal MLP Neural Network Classifier for Fault Detection of Three Phase Induction Motor. Expert Systems with Applications. 2010. vol. 37. no. 4. pp. 3468–3481. DOI: 10.1016/j.eswa.2009.10.041.
46. Teler K., Skowron M., Orłowska-Kowalska T. Implementation of MLP-Based Classifier of Current Sensor Faults in Vector-Controlled Induction Motor Drive. IEEE Transactions on Industrial Informatics. 2024. vol. 20. no. 4. pp. 5702–5713. DOI: 10.1109/TII.2023.3336348.
47. Lin W., Yang C., Lin J., Tsay M.A. Fault Classification Method by RBF Neural Network with OLS Learning Procedure. IEEE Power Engineering Review. 2001. vol. 21. no. 8. DOI: 10.1109/MPER.2001.4311561.
48. Yang P., Wang T., Yang H., Meng C., Zhang H., Cheng L. The Performance of Electronic Current Transformer Fault Diagnosis Model: Using an Improved Whale

Optimization Algorithm and RBF Neural Network. Electronics. 2023. vol. 12. no. 4. DOI: 10.3390/electronics12041066.
49. Jeong K., Choi S.B., Choi H. Sensor Fault Detection and Isolation Using a Support Vector Machine for Vehicle Suspension Systems. IEEE Transactions on Vehicular Technology. 2020. vol. 69. no. 4. pp. 3852–3863. DOI: 10.1109/TVT.2020.2977353.
50. Ramdani O., Beddek K., Haddouche R., Zerrougui M., Chouider N. SVM-Based Approach Fault Detection for PMSG-Wind Energy Conversion System. Journal of Engineering Research. 2025. vol. 13. no. 4. pp. 3378–3393. DOI: 10.1016/j.jer.2025.01.001.
51. Zhou N., Shao Q., Zhou J., Changjie H. Fault Classification of Proton Exchange Membrane Fuel Cells for Vehicles based on XGBoost. Proceedings of 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2021. pp. 1054–1058. DOI: 10.1109/ICBAIE52039.2021.9389943.
52. Hasan S., Toufikuzzaman M. Fault Occurrence Detection and Classification of Fault Type in Electrical Power Transmission Line with Machine Learning Algorithms. International Journal on Electrical Engineering and Informatics. 2022. vol. 14. no. 3. DOI: 10.15676/ijeei.2022.14.3.9.
53. Mehta A., Goyal D., Choudhary A., Pabla B.S., Belghith S. Machine Learning-Based Fault Diagnosis of Self-Aligning Bearings for Rotating Machinery Using Infrared Thermography. Mathematical Problems in Engineering. 2021. vol. 2021. no. 1. DOI: 10.1155/2021/9947300.
54. Shaikh F.A., Kamboh M.Z., Alvi B.A., Khan S., Khan F.M. Condition-Based Health Monitoring of Electrical Machines Using DWT and LDA Classifier. Sir Syed University Research Journal of Engineering and Technology. 2022. vol. 12. no. 2. pp. 95–100. DOI: 10.33317/ssurj.513.
55. Altayef E., Anayi F., Packianather M., Benmahamed Y., Kherif O. Detection and Classification of Lamination Faults in A 15 kVA Three-Phase Transformer Core using SVM, KNN and DT Algorithms. IEEE Access. 2022. vol. 10. pp. 50925–50932. DOI: 10.1109/ACCESS.2022.3174359.
56. Shinde P.V., Desavale R.G., Jadhav P.M., Sawant S.H. A Multi Fault Classification in a Rotor-Bearing System using Machine Learning Approach. Journal of the Brazilian Society of Mechanical Sciences and Engineering. 2023. vol. 45. no. 2. DOI: 10.1007/s40430-023-04015-1.
57. Rakhmawati R., Murdianto F.D., Luthfi A., Rahman A.Y. Thermal Optimization on Incubator using Fuzzy Inference System based IoT. Proceedings of International Conference of Artificial Intelligence and Information Technology (ICAIIT). IEEE, 2019. pp. 464–468. DOI: 10.1109/ICAIIT.2019.8834530.
58. Febriani A., Wahyuni R., Mardeni M., Irawan Y., Melyanti R. Improved Hybrid Machine and Deep Learning Model for Optimization of Smart Egg Incubator. Journal of Applied Data Sciences. 2024. vol. 5. no. 3. pp. 1052–1068. DOI: 10.47738/jads.v5i3.304.
59. Easwaran A., Arvindan P., Dhanyasree E., Surya R., Selvakumar S. Internet of Things Enabled Smart Animal Farm Prototype. Journal of Physics: Conference Series. 2021. vol. 2070. no. 1. DOI: 10.1088/1742-6596/2070/1/012115.
60. Guzman-Zabala J.V., Castro-Martin A.P. Smart and Semi-industrial Egg Incubator with Remote Monitoring Using LoRa Technology. Proceedings of International Conference on Computer Science, Electronics and Industrial Engineering (CSEI). Springer Nature Switzerland, 2024. pp. 522–540. DOI: 10.1007/978-3-031-70981-4_35.
61. Roshanghiyasi H., Ahmad A.H., Hosseinpour S., Jafari A., Mousazadeh H., Asadollahzadeh A. Monitoring and Controlling the Chicken Incubation Process Using

the Internet of Things System. Journal of Agricultural Mechanization. 2022. vol. 7. no. 1. pp. 73–77. DOI: 10.22034/jam.2022.15710.
62. Radhakrishnan K., Jose N., Sanjay S.G., Cherian T., Vishnu K.R. Design and Implementation of a Fully Automated Egg Incubator. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2014. vol. 3. no. 2. pp. 7686–7690.
63. Uzoigwe L.O., Ekezie J.C. Egg Incubator Control System with Short Message Service (sms) Fault Analysis Alert. Journal of Agriculture and Food Sciences. 2013. vol. 11. no. 2. pp. 45–68. DOI: 10.4314/jafs.v11i2.5.
64. Stull R. Practical Meteorology: An Algebra-based Survey of Atmospheric Science. University of British Columbia, 2017. 940 p.
65. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Representations by Back-Propagating Errors. Nature. 1986. vol. 323. no. 6088. pp. 533–536. DOI: 10.1038/323533a0.
66. Elman J.L. Finding Structure in Time. Cognitive science. 1990. vol. 14. no. 2. pp. 179–211. DOI: 10.1207/s15516709cog1402_1.
67. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural computation. 1997. vol. 9. no. 8. pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
68. Cho K., Van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014. pp. 1724–1734. DOI: 10.3115/v1/D14-1179.
69. MacKay D.J.C. Bayesian Interpolation. Neural Computation. 1992. vol. 4. no. 3. pp. 415–447. DOI: 10.1162/neco.1992.4.3.415.
70. Londhe A.S., Ingale A.S., Khadse C.B. Bayesian Regularization Neural Network-based Fault Detection System in HVDC Transmission System. Proceedings of International Conference on Smart Technologies for Energy, Environment, and Sustainable Development (ICSTEESD). Springer Nature, 2022. pp. 601–607. DOI: 10.1007/978-981-16-6875-3_48.
71. Hassan M.S., Kamal K., Ratlamwala T.A.H. Fault Classification of Power Plants using Artificial Neural Network. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. 2022. vol. 44. no. 3. pp. 7665–7680. DOI: 10.1080/15567036.2022.2113936.
72. Liu D.C., Nocedal J. On the Limited Memory BFGS Method for Large Scale Optimization. Mathematical Programming. 1989. vol. 45. no. 1. pp. 503–528. DOI: 10.1007/BF01589116
73. Ljung L. System Identification: Theory for the User. Prentice Hall, 1999. pp. 640.

Атмане Мохамед Амин — аспирант, кафедра автоматизации/факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: обнаружение и изоляция неисправностей, искусственный интеллект, автоматизация. Число научных публикаций — 1. mohamed.atmane@univ-sba.dz; PB 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +(213)4870-0138.

Затла Хишам — Ph.D., старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: автоматическое управление, искусственный интеллект, человеко-машинные системы. Число научных публикаций — 8. hicham.zatla@univ-sba.dz; PB 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +(213)4870-0138.

Толби Билал — старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: искусственный интеллект, гибридные системы, управление системами. Число научных публикаций — 15. bilal.tolbi@univ-sba.dz; РВ 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +213(4870)01-38.

Нуар Суад Фадила — старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: моделирование физических систем, численное моделирование. Число научных публикаций — 2. souadfadila.nouar@univ-sba.dz; РВ 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +213(4870)01-38.

Бухамама Мохамед — старший преподаватель, кафедра автоматизации, факультет электротехники, Джилали Лиабес Университет Сиди-Бель-Аббеса. Область научных интересов: электротехника, экологические технологии. Число научных публикаций — 14. mohammed.bouhamama@univ-sba.dz; РВ 89, 22000, Сиди Бел Аббес, Алжир; р.т.: +213(4870)01-38.

Поддержка исследований. Исследование проводится при поддержке Генерального управления по научным исследованиям и технологическому развитию (DGRSDT).