Endocrine CNN-Based Fault Detection for DC Motors

Andjela D. Djordjevic*, Miroslav B. Milovanovic, Marko T. Milojkovic, Jelena G. Petrovic, Sasa S. Nikolic
Department of Control Systems, Faculty of Electronic Engineering, University of Nis,
Aleksandra Medvedeva 14, 18104 Nis, Serbia
**andjela.djordjevic@elfak.ni.ac.rs; miroslav.b.milovanovic@elfak.ni.ac.rs; marko.milojkovic@elfak.ni.ac.rs;*
jelena.petrovic@elfak.ni.ac.rs; sasa.s.nikolic@elfak.ni.ac.rs

***Abstract*—This paper presents a novel method for detecting and classifying faults in dynamic control systems empowered with DC motors, operating under laboratory conditions. The approach employs a convolutional neural network model enhanced with an artificial endocrine influence to evaluate the condition of the rotating motor shaft by analysing information from the vibration sensors mounted on the shaft itself. The trained network effectively classifies the level of unbalance in the system into three categories based on the vibrations: optimal (no unbalance), first and second degree of unbalance. To validate the efficiency of the proposed model, its performance was compared with the performance of deep learning algorithms commonly recommended for time-series classification: default convolutional neural network, fully convolutional neural network, and residual network. The new model was shown to perform classification tasks with the highest accuracy, proving to be an efficient fault diagnosis tool with a viable potential to be applicable in industrial predictive maintenance processes.**

***Index Terms*—Convolutional neural network; Time-series classification; Artificial endocrine gland; Fault diagnosis; Vibration analysis.**

I. INTRODUCTION

In the era of rapid growth and development of artificial intelligence technologies, there is a common trend of minimising dependence on human interference for proper system functioning. This tendency is present not only because of economic benefits of the higher system autonomy but also to prevent severe damage to the system due to professional incompetence and work injuries in operating environments. Bearing that in mind, some of the latest trends in the control systems domain are affected by this urge to design autonomous systems capable of preventing the occurrence of undesirable conditions and circumstances.

Demands for system autonomy within the Industry 4.0 environment have created the growing need for the development of intelligent approaches for fault detections and disturbance classifications.

Approaches in this field are mostly based on machine learning (ML) methods that have found their massive application in the domain of system maintenance, providing the means for early fault detection, malfunction diagnosis, and maintenance scheduling based on smart prognosis to avoid unnecessary check-up costs. One of the significant advantages that intelligent malfunction diagnosis methods can provide is incipient fault detection [1]. This is possible due to the ability of intelligent models to gain insight into hidden patterns in signals by analysing previous behaviour of the system [2]. When faults are detected in their incipient stage, timely maintenance actions can prevent further fault deterioration, which could not only downgrade the system performance, but also lead to serious and irreparable damage on the system. Also, traditional diagnosis does not possess the capacity to adjust to sudden variations in the condition of the system, which can be achieved by employing intelligent diagnosis methods [3]. In addition, intelligent fault detection and diagnosis are the crucial components that form the basis of self-healing control [4]. According to the authors in [4], the self-healing methodology ensures that if a fault occurs in one part of the system, the general health and performance of the system are not affected by the given circumstances. This is achieved by selecting the right control strategy to deal with disturbances in the system. To achieve high autonomy of the system through self-healing approaches, it is imperative to implement a reliable fault diagnosis system [4]. Finally, traditional procedures that rely on the human factor are slow and often unreliable, in contrast to intelligent solutions [5].

When analysing the industrial environments and optimal operations within them, the focus should be on the vital components included in their work. One of the most common components of any industrial setup is certainly an electric motor. Recognising the importance of early detection, identification, and repair of motor faults is key to optimising the operations of the entire system. Motor fault detection can be accomplished through various signal analyses, including current, vibration, acoustic [6] and speed signal analysis [7], [8]. For example, bearing fault detection based on speed signal data acquired using a hall sensor is considered by the authors in [8]. Other research focussing on vibration analysis for fault detection can be found in [9]–[15]. The diagnosis of stator faults based on vibration signals was investigated in [9]. Papathanasopoulus, Giannousakis, Dermatas, and Mitronikas [10] considered vibration analysis for diagnosis of misalignment of brushless DC motor (BLDC) and breakdown faults. One of the drawbacks of vibration monitoring is the need to employ special sensors, which leads to an increase in

Manuscript received 22 March, 2024; accepted 12 May, 2024.

This work was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under Grant No. 451-03-65/2024-03/200102.

overall implementation costs [11]. The main issue of applying vibration analysis is not only the high cost of accelerometers, but also the fact that distinguishing the exact source of faulty vibration can be a complicated task, since the vibration signal is usually influenced by other sources in the system. In general, analysing the current signal is a cost-effective option. Stator current analysis, for instance, has been shown to provide insight into the motor health. However, it may not be as effective in detecting minor or early faults most of the time.

The discussed fault detection methodologies are strongly correlated with the maintenance approaches of the system, where many researchers focus their attention on the analysis and application of the sensed current signals for the diagnosis of malfunctions [1], [16]–[18]. To overcome characteristic issues that arise when only one type of signal is analysed, several papers discuss effective approaches for fault detection and identification based on both current and vibration signals [19]–[21]. For example, the authors in [19], [20] applied feature fusion. These approaches, where both current and vibration signals are analysed, proved to be useful for fault diagnosis as they allow the detection of mechanical and electrical defects and damages. Similarly, Suawa, Meisel, Jongmanns, Huebner, and Reichenbach [6] used vibration and sound signals for fault detection of BLDC motors. Unlike in [19], [20] where feature fusion was applied, the authors in [6] focussed on data fusion, which requires less expertise, but still improves the performance of the diagnostic algorithm. Finally, in [21], the authors considered and distinguished three health states of the motor components: healthy, incipient fault, and severe failure state. To analyse motor data, the authors applied motor current signature analysis and spectral kurtosis [21].

It is also important to note that most of the literature focussing on motor fault detection considers supervised ML models, especially neural networks [7], [15], [18]–[20], [22]. For example, in [20], Shifat and Hur did research on motor diagnosis of BLDC by applying artificial neural network (ANN) for fault classification. The authors considered both rotor and stator faults [20]. Another approach using neural networks for the diagnosis of BLDC can be found in [7], while fault classification was performed by ResNet-101 in [15]. Another interesting approach using recurrent neural network (RNN) for BLDC motor bearing fault detection and identification can be found in [19]. In [22], Abu-Rub, Ahmed, Iqbal, Toliyat, and Rahimian explored the detection of bearing fault in BLDC using the adaptive neuro fuzzy inference system (ANFIS). Similarly, Awdallah and Morcos [18] considered BLDC open-switch fault detection, where ANFIS was applied to identify and locate the faulty switch. In addition, several papers consider the application of CNNs for fault detection [1], [12], [13], [14], [17]. For example, Municoti, Das, Natarajan, and Srinivasan tackle the issue of DC motor monitoring under variable loading conditions and noise in [1]. The authors proposed two approaches using CNNs: in the first approach, feature extraction was done by convolutional layers of CNN, while the second approach considered a combination of convolutional and recurrent NNs for that task. Another approach can be found in [12], where Kim, Jung, Ko, Kong, Lee, and Youn focus on solving the malfunction diagnosis task by directly connecting different layers of the CNN network, forming a direct connection-based CNN. The application of 1-D CNNs for feature extraction and fault detection in induction motors was considered in [13], where multichannel architecture was used for vibration analysis. 1-D CNN is also applied in [14], where the focus was on unbalance detection, and Mey, Neudeck, Schneider, and Enge-Rosenblatt examined the performance of 1-D CNN, fully connected NN, hidden Markov, and random forest classifier models. Ince, Kiranyaz, Eren, Askar, and Gabbouj [17] also proposed the use of 1-D CNN as a tool to incorporate both feature extraction and classification problems into a ML model for induction motor diagnosis.

Taking into account the proven efficacy of CNNs for the diagnosis of malfunction, this research aims to design an enhanced CNN network by incorporating artificial endocrine glands [23]–[31] within its structure. Implementing an endocrine mechanism in the neural network structure enables a better response to the time-varying operating conditions, and, therefore, provides improved model performance. By producing the right amount of hormones, artificial glands influence the output of the network depending on the variations in the selected stimulus. This quality can be particularly useful for fault diagnosis. The new endocrine convolutional neural network (ECNN) model was used for the detection and classification of unbalance on the rotating shaft of the DC motor based on the vibration signal. The performance of the proposed model was compared with the performance of the traditional CNN, fully convolutional network (FCN), and ResNet models.

This article is organised as follows. In Section II, a brief overview of popular neural network approaches for time-series data classification is given. The new ECNN architecture is introduced in Section III, where its structure and way of functioning are described in detail. The data set on which the simulations were performed is described in Section IV, whereas the simulation results are presented in Section V. In Section VI, the simulation results are discussed, and a comparison of the performances of different models is made. The concluding remarks are given in the last section, along with the discussion of future research directions.

II. NEURAL NETWORK APPROACHES FOR CLASSIFICATION OF TIME-SERIES DATA

Deep learning has proven to be an effective tool for addressing time-series classification problems [32]–[34]. CNNs are frequently proposed for fault detection tasks and time-series classification processes [1], [6], [12]–[14], [17]. Such neural networks are suitable for the classification of large-scale time-series data characterised by high dimensionality. The main characteristic of CNNs lies in the fact that they consist of three types of layers, convolutional, pooling layers, and fully connected layers, where a data vector is expected as an input. Using a CNN reduces the number of parameters of the learning model and the possibility of overfitting [35]–[37].

The CNN model was one of the networks used by May, Neudeck, Schneider, and Enge-Rosenblatt in [14] and proved to perform time-series classification successfully in the DC motor vibration data set. Therefore, it was also used for the time-series classification task in this study. Although the authors in [14] used the model only to determine the absence or presence of a fault, it is essential to emphasise that the network's primary function in this research was to detect whether the system runs normally or to distinguish between two degrees of unbalance: the first degree where corrective control actions would be required, or the second degree, i.e., severe unbalance which requires immediate shutdown of the system and repair.

Also, Fawaz, Forestier, Weber, Idoumghar, and Muller [32] performed a comparative analysis of the performances of the most commonly used neural network structures, from simple ones such as multilayer perceptrons (MLPs) to more complex structures such as CNNs, echo state networks (ESNs), time-wrapping invariant ESNs (TWIESNs), fully convolutional networks (FCNs), multiscale CNNs (MCNNs), time Le-Net (t-LeNet), multichannel deep CNNs (MCDCNNs), time-CNNs, residual networks (ResNets), and encoders for time-series classification. The results of their research showed that FCN and ResNet architectures stood out in comparison to other approaches and have proven to be the most efficient tools when it comes to time-series classification. FCNs were first used for semantic segmentation in the work by Long, Shelhamer, and Darrell [33], while Wang, Yan, and Oates [34] proposed the application of FCNs for time-series classification. For this purpose, FCNs are used for feature extraction. According to the authors in [34], the most significant characteristic of FCN lies in the fact that there are no local pulling layers in the network structure. Instead, the global average pooling layer is applied to the output of the last convolution block. Therefore, there are no changes in the length of a time-series through the convolution layers [32]. The FCN architecture proposed in [34] was used in the simulations for this research. The structure consists of three convolutional layers followed by a global pooling layer and a *softmax* layer. The convolutional layer is the first layer of each convolution block. Its outputs are fed into the batch normalisation layer, which is followed by the rectified linear unit (ReLU) activation layer [34]. The FCN architecture is presented in Fig. 1. [34].

The ResNet structure is somewhat similar to the FCN, though much deeper, since no local pooling layers are implemented. It consists of three residual blocks, each containing three convolutional blocks described previously [32]. The outputs of the last block are fed into the global average layer followed by a *softmax* layer. The network outputs are equal to the number of target value classes. ResNet's most distinctive trait lies in the fact that shortcut connections, linking the input and output, are added in each residual block. In this way, the effect of the vanishing gradient is weakened [32]. The idea behind this network is to enable adaptation to the residual mapping rather than directly learning the base mapping. This approach facilitates the optimisation and training of extremely deep networks by focussing on learning residual functions. The structure proposed in [34] was applied for the classification task in this study as part of the *Simulations* section (Section V) and is presented in Fig. 2.

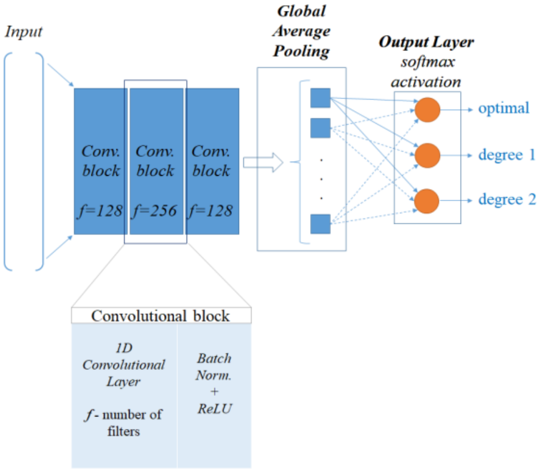

Fig. 1. FCN structure.

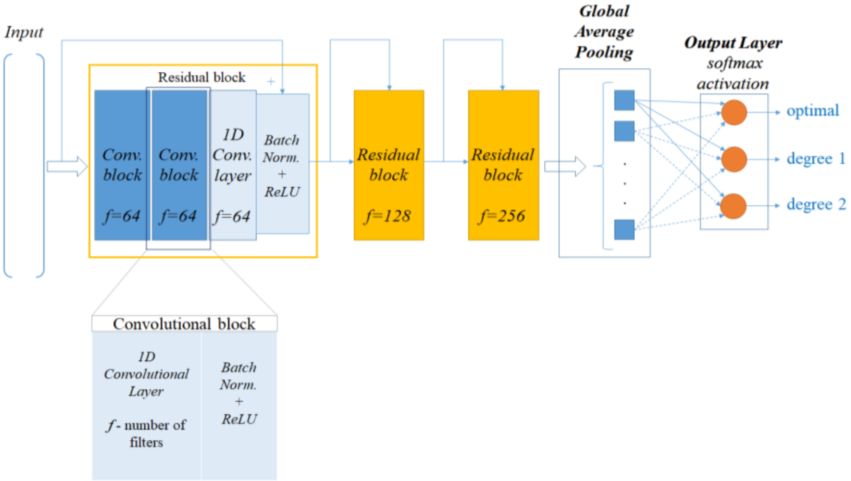

Fig. 2. ResNet structure.

Considering that CNN, FCN, and ResNet architectures have been shown to possess characteristics of crucial importance when it comes to time-series classification, it can be concluded that these models are appropriate benchmark models when it comes to evaluating the performance of new neural network models for some specific classification task. Taking into account that, in this study, we compared the performance of the traditional CNN [14], FCN [34], and ResNet [34] models for the classification of complex time-series data with the performance of the newly proposed endocrine convolutional neural network (ECNN) which will be explained in detail in Section III.

III. Design of Endocrine Convolutional Neural Network

The implementation and design of neuroendocrine models and the concept of applying endocrine mechanisms in soft computing were inspired by the way biological neural and endocrine systems interact to regulate hormonal status in blood and, therefore, maintain optimal functioning of the human body [23]–[31]. Just as the endocrine system regulates biological processes in the body by producing a required concentration of hormones as an answer to received information, endocrine neural networks employ artificial glands to incorporate this mechanism and adapt to time-varying behaviours or operating conditions [29]. Such a network contains one or more glands that react to changes and disturbances present in the system [31]. Depending on the signal received, the glands produce an adequate hormonal response in the form of a numerical value. The glands output or endocrine factors are used as external parameters commonly incorporated into one of the layers of the network, where they are applied to achieve hormonal influence [27] affecting the appropriate coefficients of network weights [28].

The novel ECNN structure is presented in Fig. 3 and possesses an improved learning ability compared to traditional CNNs by incorporating an environmental stimulus δ for activating artificial glands. The manner of work of ECNN will be described through the following subsections.

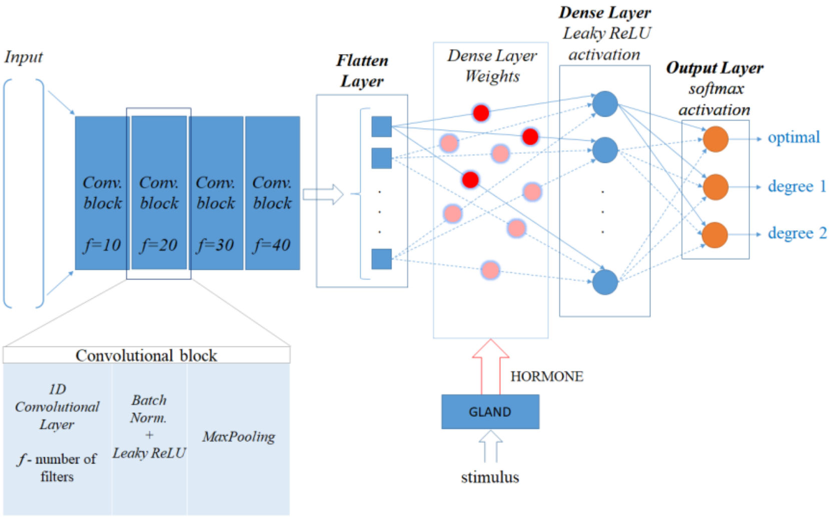

Fig. 3. ECNN structure.

A. CNN Foundation

The CNN structure employed for the classification of DC motor vibration data by the authors in [14] was used as the base for the ECNN network proposed in this study. The input size of the network corresponds to the window length, which aligns with the data collected over one second of recording. The network architecture comprises an input layer followed by a series of convolutional blocks. Each block structure commences with a convolutional layer. The output values from the convolutional layer underwent batch normalisation and served as input signals to the leaky rectified units. To down-sample the input representation, a max-pooling layer was employed as the final layer within each block. Following the convolutional blocks, two fully connected layers were incorporated with rectified linear units (ReLU) applied between them. The second fully connected layer represents the output layer. To mitigate overfitting, the dropout technique was implemented during the training process, featuring a dropout layer placed before the last fully connected layer. Finally, the network has three outputs with integrated *softmax* activation functions.

B. Integration of Endocrine Components

An endocrine factor is introduced into the network in the form of environmental stimuli (δ_1, δ_2, …, δ_i) where each stimulus causes hormonal influence on a network by introducing a proper hormone concentration value into the weights of the network. The hormone concentration of a single gland (C_g) could be expressed as [25]

$$C_g(t+1) = \beta_g C_g(t) + R_g(t+1), \tag{1}$$

where β_g is the decay constant, R_g is the stimulation parameter, and the index g represents a specific gland. The stimulation parameter can be calculated from

$$R_g(t) = \frac{\alpha_g}{1+C_g(t-1)} \sum_j \omega_{ij}(t) X_{ij}, \tag{2}$$

where α_g is the stimulation rate, ω_{ij} is the weight coefficient, and X_{ij} is the input. In addition, index i represents the current input of the network, while j is the current hormone. Finally, the output of the neuron *f(u)* can be presented as [23]

$$f\left(u\right)=f\left(\sum_{i=1}^{n}\omega_i X_i C_g S_j\right). \tag{3}$$

S_j of (3) is the hormone sensitivity parameter valued in the range between 0 and 1, and *n* is the number of neuron inputs. It could be considered that neurons with a sensitivity value close to zero are negligible. In contrast, neurons with a sensitivity value close to 1 will have a significant influence on network calculations. At its core, a sensitivity parameter S_j defines the influence of a stimulus, which is achieved using a specific number of identical artificial glands that are in charge of regulating hormone concentrations C_g. This number of glands is determined according to the requirements of the specific use case, applying the trial-and-error method. Commonly, the first trial assumes that one gland is used per stimulus, increasing the number of glands linearly until the performances improve with a significant ratio. The artificial endocrine mechanism of weight update is presented in Fig. 4.

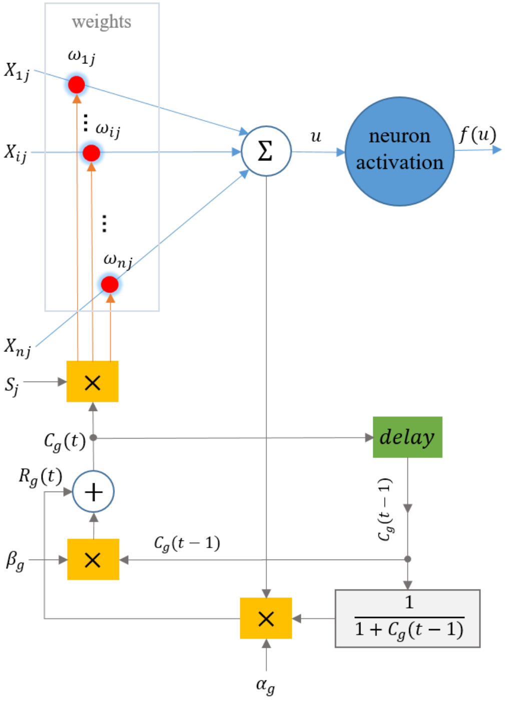

Fig. 4. Artificial endocrine mechanism of weight update.

The proposed ECNN network is based on the adaptive learning algorithm that operates using the gradient descent method [38]. It is more active when external disturbances and output signal deviations are strong. When the influence of external disturbances is minimised to a specific range, the weights become less sensitive to external factors. In general, adaptive components are provided by the introduction of hormonal influence to the weights of the ECNN dense layer network, where the approximation error *E* of the ECNN can be calculated as

$$E=y-y^{net}=(W-WC_gS_j)^T, \tag{4}$$

where $W = [w_1, w_2, \ldots, w_n]^T$ represents a vector with weight reference values, $W = \left[w_1, w_2, \ldots, w_n\right]^T$ represents a vector that contains real weights values, y is the output signal of the system sensed, and y^{net} is the output signal of the system approximated by ECNN.

C. *Neural Network Setup*

1. Selection of the environmental stimulus. ECNN environmental stimulus is selected as the motor speed (δ_1). The recorded speed is presented in Fig. 5. Due to significant computation requirements and speed process issues, a problem of insufficient memory occurred. For this reason, the mean value of the speed signal over one second is calculated for each input window.

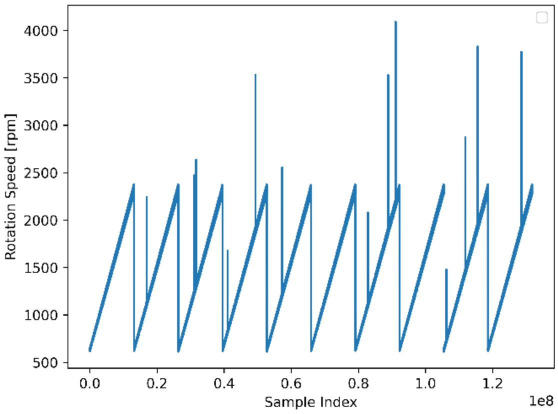

Fig. 5. Speed data for the examined data set.

2. Glands number. The trial began with two artificial glands, and the number of glands increased by one in each subsequent trial. The networks were empirically tested and structures with six glands for the specified stimulus were selected.
3. Artificial endocrine parameters. The stimulation parameter (R_g) and the hormone concentration (C_g) are determined according to [24]. Furthermore, the sensitivity parameter (S_g) was adjusted according to the following rule:

$$\frac{current\ speed}{4000}=S_g. \tag{5}$$

The value 4000 was selected from available speed data information, where it is shown that the maximal value of speed is 4000 rounds per minute (rpm). Additionally, the values of the average speed points are in the range of 0 rpm to 2500 rpm, implying that the recorded values of 2500 to 4000 represent outliers caused by poor sensing or recorded disturbances. Finally, the S_g parameter will be lower when disturbances are not recorded, reducing the influence of endocrine components on the network while the motor operates optimally. In contrast, larger S_g values (from 0.625 to 1) will be used when disturbances are sensed, influencing a larger amount of hormonal influence on the network.
4. ECNN inputs. An input to each network is the vibration signal with 4096 consecutive samples (corresponding to the window length) coming from the vibration sensor.
5. ECNN outputs. The network has three outputs corresponding to the appropriate health states of the

system, i.e., optimal system performance, faulty performance where corrective actions are needed, and deteriorating state where immediate system shutdown is required along with appropriate maintenance actions.

6. ECNN: Weights per gland. 128 neurons in the dense layer are specified. Since the number of outputs of the flatten layer is 9800, and the number of glands is six, that means four glands will provoke 209070 neuron weights and the following two glands will make an impact on 209080 neuron weights. In this way, a balanced influence of the glands on ECNN will be achieved.

IV. USE CASE

To facilitate the simulations, the data set authored by Mey, Neudeck, Schneider, and Enge-Rosenblatt [14], which is accessible via the Fraunhofer Fordatis database [39], was used. This data set encompasses information obtained from three vibration sensors placed on a rotating shaft. Data were collected under varying rotational speeds and subjected to four distinct levels of unbalance, as well as under unbalance-free conditions. Data acquisition was performed using an electronically commutated DC motor that powers a 12 mm diameter shaft. This shaft is interconnected with another shaft of identical dimensions, measuring 75 mm in length, through a coupling mechanism. The latter shaft traverses a roller bearing, secured within a bearing block, with the unbalance holder affixed directly behind it. Notably, the unbalance holder, fabricated by 3D printing, comprises a disc featuring axially symmetric recesses into which weights were inserted to replicate unbalance effects. Vibration sensors were affixed to both the bearing block and the motor mounting, with data acquisition conducted using a four-channel system.

The data sampling rate was set at 4096 values per second. The data set was partitioned into five pairs of smaller data sets, each pair housing data recorded under varying unbalance strengths. Each pair comprises a development data set and an evaluation data set. It is important to note that after the recording of the development data set, the measurement setup was completely reassembled before the recording of the evaluation data set to ensure the diversity between the two data sets [14]. The first pair of data was recorded during the normal functioning of the system, i.e., without unbalance influence. The second, third, fourth, and fifth data pairs were recorded under the influence of different unbalance sizes with 45.9 ± 1.4 mm g, 60.7 ± 1.9 mm g, 75.5 ± 2.3 mm g, and 152.1 ± 2.3 mm g unbalance factors, respectively [14].

For the development data set, data acquisition entailed increasing the motor voltage from 2.0 V to 10.05 V in increments of 0.05 V. Simultaneously, the rotational speed ranged from 630 rpm to 2330 rpm. Conversely, the evaluation data set was recorded with the motor voltage increased from 4.0 V to 8.1 V in 0.1 V increments, accompanied by a rotational speed range spanning from 1060 rpm to 1900 rpm. In both cases, voltage increments were introduced every 20 seconds.

V. SIMULATIONS

As stated in Section II, the efficiency of the proposed ECNN will be evaluated by comparing its performance with those of the CNN, FCN, and ResNet models for the time-series classification task on the previously described data set. All simulations were performed in Python.

A. Data Set

Due to the large size of the data set, only information from the first vibration sensor was used in this research [14]. To avoid noisy information during the warm-up phase of the system, the first 50.000 values were removed from the data set, as recommended by the authors in [14].

To better understand the data, in Fig. 6, vibration signals from the development data sets over time are shown. As it was previously stated, the signals were recorded while the motor voltage was increased from 2.0 V to 10.05 V. By analysing the graph, it can be concluded that the vibration amplitude increases with the increasing voltage and rotational speed. However, the intensity of vibration differs for each unbalance strength. The greatest difference in vibration amplitudes can be seen for voltage values between 3.5 V and 4.5 V, where the vibration measured in the normal health state system is significantly less intense compared to the vibration measured in the system with different levels of unbalances.

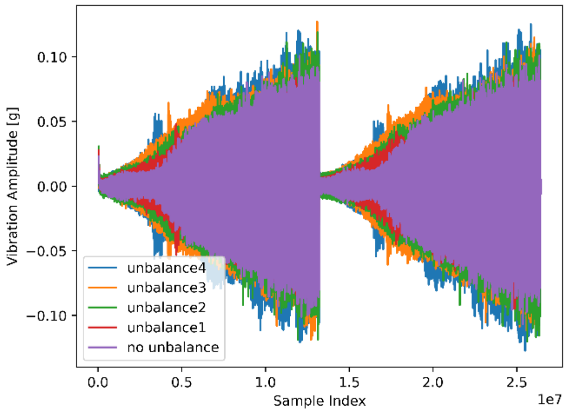

Fig. 6. Vibration signal for different levels of unbalance over time.

Table I shows the peak, root mean square value (RMS), crest factor, and standard deviation for different degrees of unbalance. The peak value represents the highest amplitude in each vibration signal and increases with the severity of the fault. This does not hold for the first degree of unbalance, where the peak value is actually smaller than in the normal state.

The RMS value is calculated using the formula

$$rms = \sqrt{\left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(a_i\right)^2}, \tag{6}$$

where sum of a_i squared refers to the sum of each squared sample of the vibration signal. A similar trend is present for the RMS value as it was for the peaks of the signals. Again, except for the first unbalance strength, the value of RMS increases with higher levels of unbalance.

The standard deviation represents the measure of distance between all values and the mean value of the vibration signal. The RMS and standard deviation values are similar, as it can be seen in Table I.

Finally, the crest factor included in Table I shows the peak to rms ratio and indicates the intensity of the peaks in the signal. The decreasing value of this factor can be an indicator of the degraded health of the component [40]. As can be seen from the table, the crest factor is mostly decreasing as the severity of the unbalance strengthens, except in the case of the third level of unbalance.

TABLE I. PEAK, RMS, CREST, AND STANDARD DEVIATION OF THE VIBRATION SIGNAL.

	Peak	RMS	Crest	Standard deviation
No unbalance	0.10675	0.008411	12.69183	0.008384
Unbalance1	0.09483	0.007540	12.57652	0.007505
Unbalance2	0.11995	0.010694	11.21702	0.010678
Unbalance3	0.12720	0.011009	11.55487	0.010987
Unbalance4	0.12719	0.013769	9.23705	0.013752

The data was then arranged into windows, each consisting of 4096 values per one second. Mey, Neudeck, Schneider, and Enge-Rosenblatt [14] conducted a binary classification of the data by marking the information from all unbalanced data sets with 1, and the information from the normal data set with 0.

In this new study, the data was grouped in a different way: three states of the systems were distinguished, and multiclass classification was performed. This way, a possibility for more precise fault diagnosis is provided. Distinguishing the intermediate state between the optimal and “problematic” system health reduces the need to immediately shut down the system every time a fault is detected, which further ensures better system productivity. The first state was marked by 0, which implies that the system runs regularly. The second state is marked with 1 and implies that the first, second, or third degree of unbalance is present. To prevent further deterioration, corrective actions are required. Finally, the third state refers to the presence of an unbalance of the fourth degree. It is marked with 2 in the data set and implies that the system needs to be shut down immediately to prevent complete system deterioration or “hazard”.

To train the models, 90 % of the data set was used as a training set, while the remaining 10 % was used for validation. This proportion proved to give better results than 80:20 or 70:30, which is shown through the trial-and-error process, and was therefore applied in this study. To test the performance of the models, the whole validation data set was used as the test set.

B. *Architectures of NNs*

In this study, the CNN architecture used in [14] was used as the initial benchmark model. As proposed in [14], the input size of the network is 4096 corresponding to the length of the window. The network architecture comprises an input layer followed by a series of convolutional blocks; concretely, in this research four blocks were applied. Each block structure starts with a convolutional layer. The number of filters was 10, 20, 30, and 40 for the first, second, third, and fourth block, respectively. The output values from the convolutional layer were normalised through batch normalisation and served as input signals to the leaky rectified units. To down-sample the input representation, a max-pooling layer was employed as the final layer within each block. Following the convolutional blocks, two fully connected layers were incorporated, with rectified linear units (ReLU) applied between them. The number of neurons in the first fully connected layer was 128 with linear activation as proposed in [14]. The output layer consisted of three neurons with a *softmax* activation function. To mitigate overfitting, the dropout technique was implemented during the training process, featuring a dropout layer placed before the last fully connected layer.

The FCN model [34] was the second model applied in the simulations. The number of inputs is the same as in the CNN model. The inputs are fed into the first convolutional block, consisting of a 1-D convolution layer with 128 filters followed by a rectified units layer with its inputs normalised using batch normalisation. The structure of the second convolution block differs in the number of filters, which is now 256. The last convolution block has the same structure as the first one. No local pooling was used between blocks. The output of the third block was fed into a 1-D global average pooling layer, followed by a fully connected layer with three output neurons and *softmax* activation [34].

The ResNet architecture proposed in [34] was also employed in this study. The model consists of three residual blocks, each consisting of three FCN convolution blocks previously described. The network consists of 4096 inputs which were normalised using batch normalisation. The number of filters in convolution layers in residual blocks is 64 for the first block and 128 for the second and third blocks. The normalised input of each residual block is fed into the rectified units layer of the last convolution block along with the regular input, enabling direct connection of the input of the residual block with the output. No local pooling layers were applied, only the global average pooling layer after the last residual block [34]. This layer is followed by a layer with three fully connected neurons with *softmax* activation.

The last model applied in the study was the proposed ECNN model. Its architecture is tailored around the basic CNN model proposed in [14] with an implemented endocrine mechanism. The main difference between the two architectures lies in the hidden dense layer, which lies between the flatten output of the last convolutional block and an output layer. In this layer, a hormonal effect is produced by multiplying the weights by a signal from the endocrine gland. The preprocessed motor speed signal in rpm was fed into the gland as a stimulus. As a consequence of that, the glands produce the right amount of “hormones”, which affects the weight values.

C. *Simulation Results*

All simulations were conducted with a Lenovo ThinkPad L14 Gen3 computer equipped with AMD Ryzen™ 5 PRO 5675U processors with integrated Intel AMD Radeon™ graphics.

1. *CNN results*

To present the simulation results obtained by CNN network, the training and validation accuracy curves are recorded and shown in Fig. 7. The validation accuracy curve is not smooth, especially in the early stages of the training. There is a certain consistency in the performance between the 45th and 55th epoch, only to drop and become unstable again until the 80th epoch is reached. The oscillations in the accuracy curve become milder after this point. From the graph, it can be concluded that the validation accuracy starts to converge after the 85th epoch, which means that more than 90 epochs are needed to stabilise the performance of the model. Taking this into account, at least 100 epochs should be executed to ensure that the accuracy of the model achieves steady state. The time needed to execute 100 epochs was around 3 hours and 40 minutes.

To evaluate the performance of the model, the test set was used further. The accuracy of the test set was 92.05647 %. The confusion matrix (Table II) shows that the class “degree 1” was misclassified the most frequently. A small number of members of this class were classified as “degree 2”. This is not as problematic as the fact that the model shows a tendency to falsely classify some members of the “degree 1” class as “optimal”, i.e., in some cases it does not recognise that a fault exists, which could lead to potential technical issues and poor predictive maintenance. In addition, the model did not falsely classify a single member of the “degree 2” class.

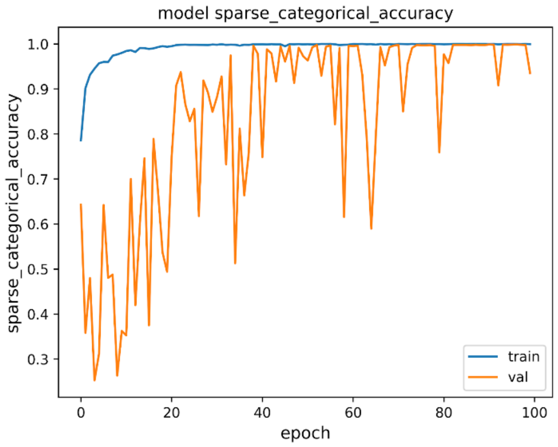

Fig. 7. CNN training and validation accuracy curves.

TABLE II. CNN CONFUSION MATRIX.

		CNN		
True	“optimal”	1665	5	0
	“degree 1”	656	4355	3
	“degree 2”	0	0	1675
		“optimal”	“degree 1”	“degree 2”
		Predicted		

2. *FCN results*

The second model used in the simulations in this study was the FCN architecture proposed in [34]. The training and validation accuracy curves are presented in Fig. 8. The training curve is smooth and stable, but this is not the case with the validation curve. The validation accuracy is pretty unstable during the training process, never reaching the training accuracy or converging even after 100 epochs. It can be noticed that the curves are apart and that 100 epochs were not enough for the proper training of the model. On the basis of the accuracy curves of the FCN model, it can be concluded that the model is not managing to capture the patterns in the data in the satisfactory manner. Also, the time needed to train the model was around 61 hours and 40 minutes, which is almost 16 times longer than the time needed to train regular CNN, where the execution of one epoch takes around 40 minutes. To optimise the process, the authors tried to use a smaller number of filters, i.e., parameters that the convolutional layer would learn from the input data, by employing a shorter window or length of the timeseries. The standard window length was equal to the number of samples collected in one second: 4096. The network was trained with 1024, 256, and 128 window lengths, but the performance of the model was severely degraded and although the training time decreased, it was still long. Bearing that in mind, the original window length was used in the end.

The accuracy recorded of FCN on the test set was 74.74578 %. By examining the confusion matrix presented in Table III, it can be concluded that, again in this case, the model shows the tendency to misclassify the members of the “degree 1” class more frequently compared to the members of the other two classes. Still, the situation is a bit different than when the CNN model is applied. The model falsely classified 24.0327 % of the members of the “degree 1” class as “optimal”, and 17.0523 % as “degree 2”, which means that 41.085 % of the members were misclassified. Regarding the classification of the “optimal” class, only a small percentage (0.8982 %) of the class was falsely classified as “degree 1”. Interestingly, the model also shows a slight tendency to mix up between the “optimal” and “degree 2” classes (1.2574 % of “optimal” was classified as “degree 2”, and 0.8955 % of “degree 2” as “optimal”, which was not the case with the other models. The classification of faulty behaviour as non-faulty presents a greater issue, since not taking adequate actions could lead to further deterioration of the system health.

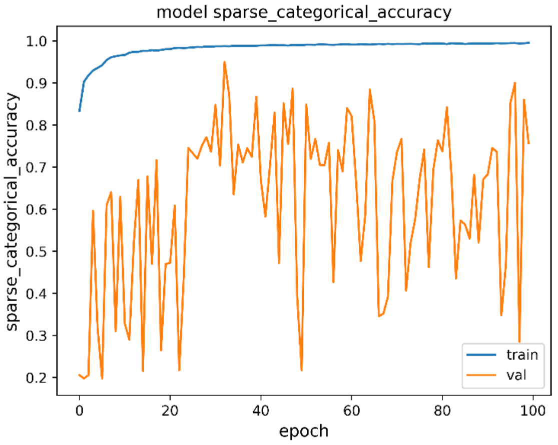

Fig. 8. FCN training and validation accuracy curves.

TABLE III. FCN CONFUSION MATRIX.

		FCN		
True	“optimal”	1634	15	21
	“degree 1”	1205	2954	855
	“degree 2”	15	0	1660
		“optimal”	“degree 1”	“degree 2”
		Predicted		

3. *ResNet results*

The training and validation accuracy curves are presented in Fig. 9. Again, the training curve is smooth, stable, and converges to 1. The validation curve is not as smooth as the training curve, but is still more stable than CNN and FCN validation curves. The curve is noisy for the first thirty epochs and starts to converge from that point. Still, there is a significant decrease in the validation accuracy around the 45th epoch. Between the 40th and 50th epoch there are oscillations in the model performance, but the validation accuracy becomes steady and stable again after the 50th epoch. There are only slight variations in the model accuracy from this point until the end of the training. However, a great drawback of the ResNet approach is the long training time. Training with 100 epochs took 144 hours or 6 days due to the complexity of the network.

From the accuracy curves, it can be concluded that the validation accuracy converges after 50 epochs. Although the validation accuracy at the end of the training is 99.97 % and 99.91 % on the training and validation sets, respectively, the model fails to adequately classify the data on the test set and has poor test accuracy: 59.98335 %. This leads to the conclusion that the ResNet model trained with 100 epochs shows signs of overfitting. To address this issue, the training was stopped after 35 (the accuracy graphs for 35 epochs of training are shown in Fig. 9) and after 60 epochs. In both cases, the accuracy was the same as that obtained after 100 epochs. To find the cause of the low accuracy, the confusion matrix was examined. Table IV shows the confusion matrix. It can be concluded that the model classifies all samples as "degree 1" class, which was the dominant class in the training set. This is a huge problem, especially since the model does not distinguish between the "degree 1" state that requires corrective actions and the "degree 2" state that requires immediate shutdown of the system.

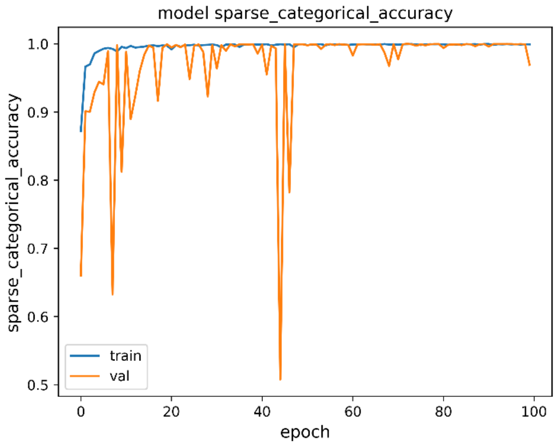

Fig. 9. ResNet training and validation accuracy curves.

TABLE IV. RESNET CONFUSION MATRIX.

		ResNet		
True	"optimal"	0	1670	0
	"degree 1"	0	5014	0
	"degree 2"	0	1675	0
		"optimal"	"degree 1"	"degree 2"
		Predicted		

4. *ECNN results*

The execution for 100 epochs of the ECNN model takes almost the same time as CNN: 3 hours and 48 minutes, but it shows slightly better performance. The training and validation accuracy curves are presented in Fig. 10. The accuracy curve shows great oscillations in the performance on the validation set during the first 60 epochs of training. After that point, the curve is much smoother and starts to converge after 75 epochs. By the end of the training, the performance is stable, and the validation curve shows high accuracy values. The accuracy in the test set was 95.80093 %, which is the best accuracy compared to the other models applied in this research.

The related confusion matrix is presented in Table V. Similar to the CNN model, the class "degree 1" was misclassified more frequently, but to a lower extent than in the CNN case. A slightly larger number of members of this class was classified as "degree 2" than during the CNN simulation. However, the number of falsely classified members of the "degree 1" class as "optimal" was significantly lower, which shows a great advantage of ECNN compared to CNN, as this type of misclassification could cause greater problems as the model does not recognise that a fault exists and the system continues to work as if its health state is optimal. The model did not falsely classify a single member of the class of "degree 2".

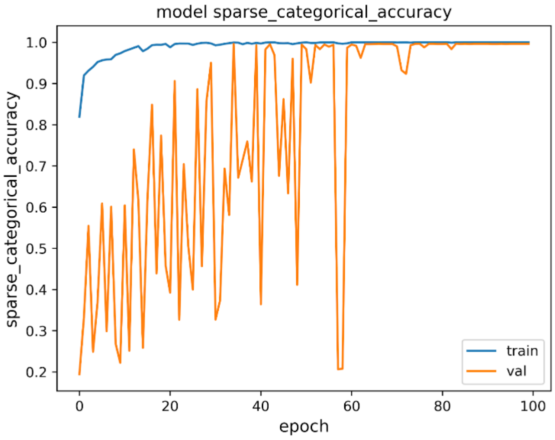

Fig. 10. ECNN training and validation accuracy curves.

TABLE V. ECNN CONFUSION MATRIX.

		ECNN		
True	"optimal"	1656	13	1
	"degree 1"	302	4677	35
	"degree 2"	0	0	1675
		"optimal"	"degree 1"	"degree 2"
		Predicted		

VI. DISCUSSION

To summarise, Table VI shows the test accuracies of each model after 100 epochs of training, and the time needed to execute the training. It is obvious that the CNN and ECNN models have achieved a much higher accuracy than FCN and ResNet. Also, the time needed to train CNN and ECNN models is significantly shorter than the time needed for FCN and ResNet models.

TABLE VI. ECNN CONFUSION MATRIX.

Model	Accuracy	Time
CNN	0.920565	3 h and 40 min
FCN	0.747458	61 h and 67 min
ResNet	0.599833	144 h
ECNN	0.958009	3 h and 48 min

The ResNet model has achieved the lowest accuracy and is unable to perform classification on the test set due to overfitting issues. The FCN model achieves better accuracy from ResNet, but still fails to perform the classification task in a satisfactory manner after 100 epochs of training. The CNN model achieves a high accuracy of 92.0565 % on the test set, while the ECNN model shows slightly better performance with 95.8009 % test accuracy.

The time needed to run CNN and ECNN training was similar, with the latter being slower. The CNN network took the least time to train, while the FCN training took a considerably longer time period to train. The ResNet training lasted the longest. The main advantage of the ECNN network in comparison to the regular CNN network lies in the fact that better performance and higher test accuracy are achieved with an insignificant difference in the time needed to train this model compared to CNN. Nevertheless, there are several drawbacks regarding the ECNN model development process. The choice of endocrine factor can have a significant influence on the behaviour of the model; thus, its performance could be drastically degraded if this factor is not selected properly. Furthermore, since there is no methodology that defines the procedures that ensure the right choice of the hyperparameter of the environmental stimulus, the selection of endocrine factors is based on the trial-and-error process. Although the ECNN training phase takes almost the same amount of time as the CNN training phase, the trial-and-error process of the factor selection is often time-consuming. Therefore, designing a reliable ECNN model can take much more time compared to the traditional CNN model, but this cannot be concluded solely on the basis of the analysis of the research results.

VII. CONCLUSIONS

The rapid growth and development of AI technologies generates a tendency to move toward sophisticated maintenance procedures in industrial systems. This inclination has an impact on the most recent advances in fault diagnosis, leading to strong demands for the design of highly intelligent strategies enabling the implementation/application of predictive maintenance in industry to reduce the requirement for hiring specialised staff.

This paper provides a contribution in the domain of fault diagnosis, proposing a novel intelligent algorithm for fault detection in DC motors. In this study, the level of unbalance of a rotating DC motor shaft is determined using an intelligent diagnostic system. The proposed diagnostic system consists of a novel neuroendocrine model: endocrine convolutional neural network (ECNN) combining traditional CNN structure with artificial endocrine influence through artificial glands. The efficiency of the designed model was evaluated by comparing its performance with those of the CNN, FCN, and ResNet models for the complex time-series classification of the vibration signal. Tracking and analysing vibration signals is of significant importance for a proper predictive maintenance of equipment in industrial environments, which motivated the authors to verify the efficiency of ECNN using this signal.

The model has been shown to achieve a high accuracy of 95.8009 % with a considerably shorter training time in comparison to the ResNet and FCN models. Its main advantage compared to the traditional CNN lies in the higher accuracy obtained with a small/negligible difference in training time. With this in mind, the newly proposed ECNN model can be used and implemented successfully as a part of the diagnostic system. However, it should be noted that, compared to the traditional CNN model, developing a reliable ECNN model consumes considerably longer time considering not only the training time, but also the time needed to select endocrine factors.

The development of a reliable neuroendocrine diagnostic system, along with its implementation for predictive maintenance and self-healing control, represents the main goal/course of our future research. The concept behind self-healing control lies in the idea that the functionality of the entire system remains unaffected by malfunction in a single part of the system by choosing an adequate control strategy based on the signal from the diagnostic system. This method gives opportunity to respond to the demands for greater autonomy of the system. The research efforts will be orientated towards 3D crane, ABS system, servo systems, and two rotor aerodynamic systems.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] S. Munikoti, L. Das, B. Natarajan, and B. Srinivasan, "Data-driven approaches for diagnosis of incipient faults in DC motors", *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5299–5308, 2019. DOI: 10.1109/TII.2019.2895132.

[2] S. Qiu, X. Cui, Z. Ping, N. Shan, Z. Li, X. Bao, and X. Xu, "Deep learning techniques in intelligent fault diagnosis and prognosis for industrial systems: A review", *Sensors*, vol. 23, no 3, p. 1305, 2023. DOI: 10.3390/s23031305.

[3] G. Gao, D. Zhou, H. Tang, and X. Hu, "An intelligent health diagnosis and maintenance decision-making approach in smart manufacturing", *Reliability Engineering & System Safety*, vol. 216, art. 107965, 2021. DOI: 10.1016/j.ress.2021.107965.

[4] H. Liang and X. Yin, "Self-healing control: Review, framework, and prospect", *IEEE Access*, vol. 11, pp. 79495–79512, 2023. DOI: 10.1109/ACCESS.2023.3298554.

[5] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals", *Sensors*, vol. 17, no. 2, p. 425, 2017. DOI: 10.3390/s17020425.

[6] P. Suawa, T. Meisel, M. Jongmanns, M. Huebner, and M. Reichenbach, "Modeling and fault detection of brushless direct current motor by deep learning sensor data fusion", *Sensors*, vol. 22, no. 9, p. 3516, 2022. DOI: 10.3390/s22093516.

[7] C.-Y. Lee, C.-H. Hung, and T.-A. Le, "Intelligent fault diagnosis for BLDC with incorporating accuracy and false negative rate in feature selection optimization", *IEEE Access*, vol. 10, pp. 69939–69949, 2022. DOI: 10.1109/ACCESS.2022.3186753.

[8] O. Zandi and J. Poshtan, "Fault diagnosis of brushless DC motors using built-in hall sensors", *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8183–8190, 2019. DOI: 10.1109/JSEN.2019.2917847.

[9] T. Zimnickas, J. Vanagas, K. Dambrauskas, A. Kalvaitis, and M. Ažubalis, "Application of advanced vibration monitoring systems and long short-term memory networks for brushless DC motor stator fault monitoring and classification", *Energies*, vol. 13, no. 4, p. 820, 2020. DOI: 10.3390/en13040820.

[10] D. A. Papathanasopoulos, K. N. Giannousakis, E. S. Dermatas, and E. D. Mitronikas, "Vibration monitoring for position sensor fault diagnosis in brushless DC motor drives", *Energies*, vol. 14 no. 8, p. 2248, 2021. DOI: 10.3390/en14082248.

[11] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders", *IEEE/CAA Journal of Automatica Sinica*, vol. 6 no. 2, pp. 441–451, 2019. DOI: 10.1109/JAS.2019.1911393.

[12] M. Kim, J. H. Jung, J. U. Ko, H. B. Kong, J. Lee, and B. D. Youn, "Direct connection-based convolutional neural network (DC-CNN) for fault diagnosis of rotor systems", *IEEE Access*, vol. 8, pp. 172043–172056, 2020. DOI: 10.1109/ACCESS.2020.3024544.

[13] I. H. Ozcan, O. C. Devecioglu, T. Ince, L. Eren, and M. Askar, "Enhanced bearing fault detection using multichannel, multilevel 1D CNN classifier", *Electrical Engineering*, vol. 104, no. 2, pp. 435–447, 2022. DOI: 10.1007/s00202-021-01309-2.

[14] O. Mey, W. Neudeck, A. Schneider, and O. Enge-Rosenblatt, "Machine learning-based unbalance detection of a rotating shaft using vibration data", in *Proc. of 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, pp. 1610–1617. DOI: 10.1109/ETFA46521.2020.9212000.

[15] S.-L. Lin, "Application combining VMD and ResNet101 in intelligent diagnosis of motor faults", *Sensors*, vol. 21, no. 18, p. 6065, 2021. DOI: 10.3390/s21186065.

[16] R. L. V. Medeiros, A. C. L. Filho, J. G. G. S. Ramos, T. P. Nascimento, and A. V. Brito, "A novel approach for speed and failure detection in brushless DC motors based on chaos", *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8751–8759, 2019. DOI: 10.1109/TIE.2018.2886766.

[17] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time

motor fault detection by 1-D convolutional neural networks”, *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016. DOI: 10.1109/TIE.2016.2582729.

[18] M. A. Awadallah and M. M. Morcos, “Automatic diagnosis and location of open-switch fault in brushless DC motor drives using wavelets and neuro-fuzzy systems”, *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 104–111, 2006. DOI: 10.1109/TEC.2004.841502.

[19] W. Abed, S. Sharma, R. Sutton, and A. Motwani, “A robust bearing fault detection and diagnosis technique for brushless DC motors under non-stationary operating conditions”, *Journal of Control, Automation and Electrical Systems*, vol. 26, pp. 241–254, 2015. DOI: 10.1007/s40313-015-0173-7.

[20] T. A. Shifat and J.W. Hur, “ANN assisted multi sensor information fusion for BLDC motor fault diagnosis”, *IEEE Access*, vol. 9, pp. 9429–9441, 2021. DOI: 10.1109/ACCESS.2021.3050243.

[21] T. A. Shifat and J. W. Hur, “An effective stator fault diagnosis framework of BLDC motor based on vibration and current signals”, *IEEE Access*, vol. 8, pp. 106968–106981, 2020. DOI: 10.1109/ACCESS.2020.3000856.

[22] H. Abu-Rub, S. M. Ahmed, A. Iqbal, H. A. Toliyat, and M. M. Rahimian, “Incipient bearing fault detection for three-phase brushless DC motor drive using ANFIS”, in *Proc. of 8th IEEE Symposium on Diagnostics for Electrical Machines*, *Power Electronics & Drives*, 2011, pp. 620–625. DOI: 10.1109/DEMPED.2011.6063688.

[23] C. Sauze and M. Neal, “Artificial endocrine controller for power management in robotic systems”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 1973–1985, 2013. DOI: 10.1109/TNNLS.2013.2271094.

[24] J. Timmis, M. Neal, and J. Thorniley, “An adaptive neuro-endocrine system for robotic systems”, in *Proc. of 2009 IEEE Workshop on Robotic Intelligence in Informationally Structured Space*, 2009, pp. 129–136. DOI: 10.1109/RIISS.2009.4937917.

[25] J. Timmis, L. Murray, and M. Neal, “A neural-endocrine architecture for foraging in swarm robotic systems”, in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence*, vol. 284. Springer, Berlin, Heidelberg, 2010, pp. 319–330. DOI: 10.1007/978-3-642-12538-6_27.

[26] M. Milojković, D. Antić, M. Milovanović, S. S. Nikolić, S. Perić, and M. Almawlawe, “Modeling of dynamic systems using orthogonal endocrine adaptive neuro-fuzzy inference systems”, *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 9, p. 091013, 2015. DOI: 10.1115/1.4030758.

[27] S. L. Perić, D. S. Antić, M. B. Milovanović, D. B. Mitić, M. T. Milojković, and S. S. Nikolić, “Quasi-sliding mode control with orthogonal endocrine neural network-based estimator applied in anti-lock braking system”, *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 754–764, 2016. DOI: 10.1109/TMECH.2015.2492682.

[28] M. B. Milovanović, D. S. Antić, M. T. Milojković, S. S. Nikolić, S. L. Perić, and M. D. Spasić, “Adaptive PID control based on orthogonal endocrine neural networks”, *Neural Networks*, vol. 84, pp. 80–90, 2016. DOI: 10.1016/j.neunet.2016.08.012.

[29] M. Milovanović, D. Antić, M. Milojković, S. S. Nikolić, M. Spasić, and S. Perić, “Time series forecasting with orthogonal endocrine neural network based on postsynaptic potentials”, *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 4, p. 041006, 2017. DOI: 10.1115/1.4035090.

[30] D. Chen, J. Wang, F. Zou, W. Yuan, and W. Hou, “Time series prediction with improved neuro-endocrine model”, *Neural Computing and Applications*, vol. 24, pp. 1465–1475, 2014. DOI: 10.1007/s00521-013-1373-3.

[31] M. Milovanović, A. Oarcea, S. Nikolić, A. Djordjević, and M. Spasić, “An approach to networking a new type of artificial orthogonal glands within orthogonal endocrine neural networks”, *Applied Sciences*, vol. 12, no. 11, p. 5372, 2022. DOI: 10.3390/app12115372.

[32] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: A review”, *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019. DOI: 10.1007/s10618-019-00619-1.

[33] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation”, in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[34] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline”, in *Proc. of 2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1578–1585. DOI: 10.1109/IJCNN.2017.7966039.

[35] K. O’Shea and R. Nash, “An introduction to convolutional neural networks”, 2015. DOI: 10.48550/arXiv.1511.08458.

[36] L. Sadouk, “CNN approaches for time series classification”, in *Time Series Analysis - Data, Methods, and Applications*. IntechOpen, 2018, pp. 1–23. DOI: 10.5772/intechopen.81170.

[37] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, “Convolutional neural networks for time series classification”, *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017. DOI: 10.21629/JSEE.2017.01.18.

[38] J. Hertz, A. Krogh, R. G. Palmer, and H. Horner, “Introduction to the theory of neural computation”, *Physics Today*, vol. 44, no. 12, p. 70, 1991. DOI: 10.1063/1.2810360.

[39] O. Mey, W. Neudeck, and A. Schneider, “Vibration measurements on a rotating shaft at different unbalance strengths”, Mar. 25, 2020. [Online]. Available: https://fordatis.fraunhofer.de/handle/fordatis/151.2

[40] A. Filippenko, S. Brown, and A. Neal, U.S. Patent 6,370,957 B1, Apr. 16, 2002. [Online]. Available: https://patents.google.com/patent/US6370957B1/en