



Enhanced Fault Diagnosis in Motor Bearings: Leveraging Optimized Wavelet Transform and Non-Local Attention



Syed Ziaur Rahman¹ , Ramesh Vatambeti^{2*}

¹ Faculty of Information Technology, Majan University College (Affiliated to University of Bedfordshire, United Kingdom), 710 Muscat, Oman

² School of Computer Science and Engineering, VIT-AP University, 522237 Vijayawada, India

* Correspondence: Ramesh Vatambeti (ramesh.v@vitap.ac.in)

Received: 10-22-2023

Revised: 11-26-2023

Accepted: 12-10-2023

Citation: S. Z. Rahman and R. Vatambeti, “Enhanced fault diagnosis in motor bearings: Leveraging optimized wavelet transform and non-local attention,” *Int J. Knowl. Innov Stud.*, vol. 1, no. 2, pp. 127–138, 2023. <https://doi.org/10.56578/ijkis010205>.



© 2023 by the author(s). Published by Acadlore Publishing Services Limited, Hong Kong. This article is available for free download and can be reused and cited, provided that the original published version is credited, under the CC BY 4.0 license.

Abstract: Recent advancements in non-destructive testing methodologies have significantly propelled the efficiency of bearing defect detection, vital for maintaining optimal final quality standards. This study introduces a novel approach, integrating an Optimized Continuous Wavelet Transform (OCWT) and a Non-Local Convolutional Block Attention Module (NCBAM), to elevate fault diagnosis in motor bearings. The OCWT, central to this methodology, undergoes fine-tuning through a newly formulated metaheuristic algorithm, the Skill Optimization Algorithm (SOA). This algorithm bifurcates into two critical components: the acquisition of expertise (exploration) and the enhancement of individual capabilities (exploitation). The NCBAM, proposed for classification, adeptly captures long-range dependencies across spatial and channel dimensions. Furthermore, the model employs a learning matrix, adept at synthesizing spatial, channel, and temporal data, thus effectively balancing diverse data contributions by extracting intricate interrelations. The model’s efficacy is rigorously validated using a gearbox dataset and a motor bearing dataset. The outcomes reveal superior performance, with the model achieving an average accuracy of 94.17% on the bearing dataset and 95.77% on the gearbox dataset. These results demonstrably surpass those of existing alternatives, underscoring the model’s potential in enhancing fault diagnosis accuracy in motor bearings.

Keywords: Optimized Continuous Wavelet Transform (OCWT); Skill Optimization Algorithm (SOA); Non-Local Convolutional Block Attention Module (NCBAM); Fault diagnosis; Bearing faults identification

1 Introduction

Aeroengines, chemical procedures, manufacturing systems, electric machines, wind energy conversion equipment, and vehicle dynamics are only a few examples of safety-critical systems [1]. Reliability and safety are paramount in industrial systems that are vulnerable to process abnormalities and component breakdowns. By promptly identifying anomalies or mistakes, defect-tolerant operations might potentially avert performance dips and detrimental situations [2]. A flaw exists whenever a distinguishing feature or characteristic of the system deviates from its typical or anticipated behavior [3]. Disconnecting a part of the system or a sensor that isn’t functioning correctly (such as a sensor that doesn’t respond to changes in its scalar factor or is fixed at a single value) might cause issues. Actuator, sensor, and plant faults (also called component or parameter faults) are the three main categories into which these errors are often categorized [4]. A decline in system performance or even collapse can occur as a result of these failures, which either stop the controller’s action on the plant, cause major measurement errors, or directly alter the plant’s dynamic input and output attributes [5]. It is common practice to employ fault diagnostics to track out and fix problems with systems that make use of redundancy—whether it be software or hardware—in an effort to make them more reliable (also known as analytical redundancy) [6].

Attention mechanisms, often used in machine learning and artificial intelligence, allow models to focus on certain parts of input data. The “non-local” aspect suggests that the attention mechanism considers relationships between distant elements in the input, not just nearby ones. This could be particularly useful in analyzing signals or data where important information may be distributed across various locations.

Putting it all together, it seems like the approach involves using an enhanced or optimized version of the Continuous Wavelet Transform along with a non-local attention mechanism to effectively diagnose faults in motor bearings. This may be part of a broader framework that incorporates machine learning or signal processing techniques for automated fault detection in industrial systems. Instead of relying on signal processing for traditional defect detection models, intelligent diagnostic systems are used to pull out relevant features from industrial monitoring data. Extracting features, selecting features (FS), and classifying faults are the three stages that make up general intelligent diagnosis models [7, 8]. The feature extraction process begins by converting the raw data signals collected by various time- and frequency-domain sensors into trustworthy, representative features that can be used to identify faults. Second, FS excludes characteristics with lesser sensitivity and irrelevant data [9, 10]. Lastly, classification results are found through repeated training that is fed into a fault classifier by fault identification. The fault classifier then does pattern analysis [11]. Using and validating the preset algorithms reveals that their shallow network design limits their feature extraction capability and makes them difficult to implement in other applications, particularly when dealing with massive data [12, 13]. In order to compute intelligent failure diagnostics, developers now have access to unified manual feature extraction and shallow machine learning approaches [14]. Consequently, creating data-driven fault analysis models that are smart, automated, and adaptable is crucial. Recent years have seen significant techniques, making them the focus of current research for defect diagnostic approaches.

The application of DL and DNN models in mechanical defect diagnostic activities has recently attracted a great deal of interest from researchers [15]. Signal representation, feature extraction, and classification are the three steps that make up the optional model in this study. To begin, the raw vibration signals are pre-processed using the OCWT. The features are retrieved using a hexadecimal local adaptive binary pattern (HLABP). For fault classification in this study, a 33-layer convolutional neural network (CNN) architecture is suggested, followed by a non-local convolutional block module. Convolution will improve the visibility of both the spatial and channel data. Additionally, non-local attention captures the long-range interdependence of representative characteristics along the spatial and channel axes. Lastly, a learned matrix is employed to combine the spatial, channel, and temporal data. To compensate for weight, the learned matrix is designed to mine rich connection information across all three categories of data. Classification results are now easier to understand thanks to NCBAM.

Here is the breakdown of the enduring sections of the paper: The relevant literature is reviewed in Section 2, and Section 3 provides a concise clarification of the suggested model. Section 4 proves the experimental analysis, and Section 5 gives the study's conclusion.

2 Related Works

Zabin et al. [16] proposed a hybrid DTL architecture that uses 2D pictures augmented by the Hilbert transform to extract spatial and temporal data. This design consists of layers. For the experimental evaluation, we used datasets: one for industrial machinery malfunction investigations and inspections, one for toy anomaly operating sounds, and one for machinery failure prevention knowledge bearing vibration faults. These datasets were subjected to varying loads and noise levels. On the evaluated datasets, the suggested model with a 32×32 input size earned an average F1 score of 0.998. The suggested model achieved its maximum accuracy, and the training epoch reduction was more than fivefold after using transfer learning with the three benchmark datasets. Also, in terms of accuracy, the suggested model was better than the state-of-the-art replicas across a range of settings.

Li and Zhao [17] offer a generic method for constructing a semantic knowledge base that gives supplementary discriminant images of various errors. The second step is to connect two variational autoencoders and make a bidirectional alignment network. This will make it possible to combine data and attributes, and describing attributes will help with finding faults. In the third place, we make the global model more generalizable and able to adapt to new generations by creating a cloud-edge cooperation model aggregation technique that uses a generative replay mechanism to include each client's knowledge. The suggested framework for unseen classes is demonstrated by experiments carried out on groups.

An extremely low-quality and noisy data environment, like a genuine factory, is no match for the lightweight model suggested by Shin and Lee [18]. Our proposed CNN-LSTM model employs short-time Fourier transform (STFT). The model is lightweight, requiring only approximately 6.6% of the CNN, and it achieves a performance difference of less than 0.5 percent, making it highly effective in terms of application.

In their study, Brusa et al. [19] sought to examine how well SHapley Additive exPlanation (SHAP) can determine which aspects are crucial for condition monitoring programs for rotating equipment to use in detecting and classifying faults. In this case study, the writers focus on medium-sized bearings that are important in industry. The mechanical engineering laboratory at Politecnico di Torino has a test rig for industrial bearings that was used to gather vibration data for various health conditions. Using SHAP, we can understand the diagnostic models. Both models attain accuracies higher than 98.5% when the SHAP is used as a feature selection criteria. We find that the most important factors influencing the results of the models are the skewness and form factor of the signal.

When it comes to training industrial fault diagnosis models, Chen et al. [20] have expertise with the Federated

Opportunistic Block Dropout (FEDOBD) method. It manages to drastically cut down on communication overhead without sacrificing model performance by breaking models into semantic chunks and letting FL players upload quantized versions of the blocks they deem important. Two coal chemical plants in two different Chinese cities have been using FEDOBD to construct industrial failure prediction models since its deployment in February 2022, when it was part of ENN Group. The model's performance remained above the 85% test F1 score, and the business was able to minimize the training communication overhead by more than 70% associated with its prior AI Engine. This is the only dropout-based FL method that we are aware of that has been effectively implemented.

To accomplish accurate compound defect diagnosis for robots, Chen et al. [21] offer an integrated solution that incorporates two small transformer networks. In this method, the six-axis industrial robot's feedback current signals are first turned into a time-frequency picture using continuous wavelet transformation (CWT). The second step in denoising the time-frequency picture is to propose a new deep learning technique, the compact Uformer. Next, a compact convolutional transformer (CCT) is used to diagnose compound faults using the denoised time-frequency pictures. This research used a collection of actual industrial robot compound faults to inform its experimental findings. Results from experiments show that, when associated with state-of-the-art algorithms, the suggested technique can achieve adequate compound defect diagnostic accuracy using data obtained from a noisy environment.

In order to better detect gearbox faults in industrial systems, Alnfai [22] developed a model called ISOSDL-FD, which combines deep learning with fault diagnostics. One of the primary goals of the proposed ISOSDL-FD technique is to detect and categorize errors in gearbox data. Also, the energy in the machinery signals may be revealed in the time-frequency representation using a fast kurtogram-based time-frequency analysis. Also used for defect detection and classification is the deep network, or DBRNN. Optimal tuning of the DL technique was finally achieved with the ISOS methodology, leading to improved classification performance. The improved performance of the ISOSDL-FD method may be demonstrated by a thorough experimental examination. The experimental findings demonstrated that the ISOSDLFD algorithm outperformed the state-of-the-art methods.

3 Proposed System

3.1 Data Gathering and Preprocessing

A variety of weights and rotational speeds determine the behavior of spinning machinery. In order to teach it to identify faults in various operational situations, it is necessary to gather the machine's vibration signal throughout the whole speed and load range [23]. However, CWTS exhibits substantial variance when the signal instance frequencies are at odds with the rotating frequencies, which can be caused by a number of different rotating rates. In order to remove these controls, the data on the rotational speed is collected with the vibration signal. Specifically, the training sample's rotational speed is taken for granted as constant because it is measurable while the machinery is in a constant state of operation. At first, we remove the DC module from the vibration signal because it doesn't offer error analysis. By subtracting the signal's mean value, a DC component is removed. Due to the fact that the rotating speed changes during start-up and shutdown, as well as during changes in the working mode and load, the CWTS effectively produces several outputs without preprocessing the signals at rotating speed. It is possible to get rid of the rotational speed control on CWTS by setting up signal re-sampling with a virtual frequency VSF. Since the training samples contain vibration signals, VSF is a frequency group consisting of q multiples of the detected spinning speed. Notably, q remains consistent with all training samples. By this re-sampled signal $x(k)$ ($k = 1, 2, \dots, m$), it is collected at frequency $f_m = n/60$. Determine f_d as the virtual frequency which is the needed numerous times of the f_m i.e., $f_d = qf_m$, in which q is the needed several statistics. For unifying the sample frequency as f_d , the processed as shadows.

By resampling frequency f_d , the k -th point must be $\bar{x}(k) = x\left(\frac{kf}{f_d}\right)$. When f is a multiple of f_d , it only requires selecting $x\left(\frac{i \times f}{f_d}\right)$ ($i = 1, 2, 3, \dots$) as the novel $\bar{x}(k)$. If not, using an interpolation operation Φ to finish the actual cases about $x\left(\left[\frac{kf}{f_d}\right]\right)$, the novel $\bar{x}(k)$ ($k = 1, 2, 3, \dots$) is accomplished by utilizing Eq. (1):

$$\begin{aligned} \bar{x}(k) &= \Phi(\vec{K}, \vec{X}) \\ \vec{K} &= \left(\left[\frac{kf}{f_d} \right] - 1, \left[\frac{kf}{f_d} \right], \left[\frac{kf}{f_d} \right] + 1, \left[\frac{kf}{f_d} \right] + 2 \right) \\ \vec{X} &= \left(x\left(\left[\frac{kf}{f_d}\right] - 1\right), x\left(\left[\frac{kf}{f_d}\right]\right), x\left(\left[\frac{kf}{f_d}\right] + 1\right), x\left(\left[\frac{kf}{f_d}\right] + 2\right) \right) \end{aligned} \quad (1)$$

Prior to processing, all data points have the same length when sampled at frequencies that are multiples of frequencies. By drawing on relatives of wavelet functions, the wavelet transforms degrade a signal in the time-frequency domain. An important wavelet function's scaling and translation are defined by:

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) \quad a, b \in \mathbb{R}, a \neq 0 \quad (2)$$

where, b is a variable, and $\psi_{a,b}(t)$ is a continuous wavelet.

STFT localization was designed by CWT. As a diagnostic and processing tool, a CWT is useful for time-frequency signals. One way to get the CWT of a signal $x(t)$ is to convolution it with the wavelet function. Data is decomposed from scale 1 to l using the CWT in this method; l is often the same as or better than $2q$.

$$C_a(k) = \int x(t) \cdot \bar{\psi}_{a,b}(t) dt \quad (3)$$

where, C_a (a is identified by SOA) is the wavelet constant of $x(t)$ at the a -th scale and $\bar{\psi}_{a,b}(t)$ is the problematic conjugate of translation b . Coefficients for the various signal components are generated by the CWT using a number of scaling factors. A two-dimensional picture is rendered directly from a signal domain by means of this wavelet coefficient. The CWTS is produced by graphing the wavelet coefficients.

To get every wavelet coefficient in a matrix $P = [C_1, C_2, \dots, C_l]$, it is altered to a gray matrix P_{new} by:

$$P_{\text{new}}(i, j) = \left\lfloor \frac{P(i, j) - p_{\min}}{p_{\max} - p_{\min}} \times 255 + \frac{1}{2} \right\rfloor \quad (4)$$

where, p_{\max} and p_{\min} are the rudiments of P , congruently. A value of the component in P_{new} refers to a gray charge in the sequence from 0 to 255. So, P_{new} is CWTS of the unique signal.

3.2 SOA

This section introduces the suggested SOA and presents its mathematical modeling for detecting the value of a . Members of SOA, a population-based technique, are people just like you and me, always looking for ways to do better at what we do. Members of the SOA population are, in reality, potential answers to the optimization issue at hand. These members' locations in the search space represent the difficulty's variables' values. At the start of the procedure, the sites of members of the SOA are initialized randomly. In accordance with Eq. (5), a matrix may be used to mathematically represent the SOA population.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \quad (5)$$

The i -th candidate solution is denoted by X_i ; in this case, the N th member of the SOA proposes the value of the d -th variable, m is the sum of variables, and N is the sum of members of the SOA.

Everyone in the population might be the key that unlocks the problem's solution. To rephrase, the objective function is assessed by assigning each member to a variable. As a result, the goal function standards may be characterized mathematically by a vector in accordance with Eq. (6).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \quad (6)$$

In this case, F is the function value acquired, and F_i is received from the i -th candidate solution. The best member is identified by the best value, and the worst member is identified by the worst value when seeing the standards assessed for the function. Each iteration along with the population members and objective function values, because that is how it works.

There are two steps to update members of a population in SOA: exploring and exploiting. Just as when you acquire a new ability from a pro, the discovery phase relies on mimicking that technique. During the exploitation phase, the focus is on creating an environment where individuals may practice and enhance their skills. The two-stage update process in SOA design consists of exploration, which aims to search the problem-solving space globally, and exploitation, which aims to search it locally. During the exploration phase, SOA is built in such a way that members navigate the search space guided by different members rather than being limited to only following the best member's path. The procedure's ability to precisely scan the search space and locate the initial ideal area is enhanced by this. In contrast, the algorithm converges to optimal solutions during the exploitation phase by doing local searches close to each population member.

3.2.1 Stage 1: Skill acquisition from experts

Initially, under the supervision of an established expert in the community, every SOA member sets out to learn a new talent. The value that each member of the population gets from the objective function is proportional to their quality. According to the goal function's value, the expert member has the best conditions. Watching and studying experts in action is a fundamental part of skill acquisition. This may involve observing their techniques, decision-making processes, and overall approach to the task at hand. An "experts set" is defined as all other members of a SOA that have an objective function value higher than a given member. One member of this group is chosen at random to serve as an expert trainer for the associate in question. Consequently, the chosen expert may not be the optimal choice to lead the SOA member. Actually, every single member of the SOA should make the top candidate solution a permanent part of their expertise set. By mastering the skill, the expert member guides the population members to other locations in the search space, demonstrating the algorithm's capacity for global search and exploration. If the objective function is improved by the newly computed location for every member of the population, then it is acceptable. Thus, utilizing the aforementioned principles and Eqs. (7) and (8), the first phase of the update may be represented mathematically.

$$X_i^{P1} : x_{i,d}^{p1} = x_{i,d} + r \times (E_{i,d} - I \times x_{i,d}), E_i = X_k \quad (7)$$

where, $F_k < F$ and k is arbitrarily chosen from $\{1, 2, \dots, N\}$, $k \neq i$.

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} < F_i \\ X_i, & \text{else} \end{cases} \quad (8)$$

where, X_i^{P1} is the new intended status of i^{th} applicant key based on the primary phase, $x_{i,d}^{p1}$ is its d^{th} dimension, F_i^{P1} function value, E_i is the designated expert to leader and train the i^{th} populace member, $E_{i,d}$ signifies its d^{th} sum in intermission [0-1], and I is a chance sum which is selected arbitrarily from set of $\{1, 2\}$.

3.2.2 Phase 2: Skill improvement based on practice and separate effort (exploitation)

Individual practice and activity allow each member of the population to strive for improvement in the abilities learned in the first phase of each phase. In SOA, this idea is represented by search with the goal of cumulative exploitation, wherein each member searches for better circumstances near its place to increase the charge of its objective function, which represents its degree of competence. The newly computed location in this step is considered acceptable, like in the previous phase, if it improves the charge of the goal function. Eqs. (9) and (10), which describe the notions of this phase of SOA updating theoretically, are used.

$$X_i^{P2} : x_{id}^{P2} = \begin{cases} x_{i,d} + \frac{1-2r}{t} \times x_{id}, & r < 0.5 \\ x_{i,d} + \frac{lb_j + r(ub_j - lb_j)}{t}, & \text{else} \end{cases} \quad (9)$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} < F_i \\ X_i, & \text{else} \end{cases} \quad (10)$$

where, X_i^{P2} is the new intended rank of i^{th} applicant key based on stage, x_{id}^{P2} indicates its d^{th} dimension, F_i^{P2} is its impartial purpose charge, t is the repetition counter, lb_j and ub_j are j^{th} variable.

3.2.3 Repetition procedure and pseudocode of SOA

Once all members have been efficient based on the first two phases of SOA, the first iteration is finished. The procedure is then repeated in accordance with Eqs. (7) to (10), and the algorithm moves on to the next repetition. The most optimal candidate solution is output by SOA after it is fully developed.

Algorithm 1: Pseudocode of SOA

Start SOA.

1. Input the optimization problem information.
 2. Set T (number of iterations) and N (number of population members).
 3. For $t = 1 : T$
 5. For $i = 1 : N$
 6. Phase 1:
 7. Select expert member.
 8. Calculate new status of i th candidate solution based on phase 1 using Eq. (7).
 9. Update i th candidate solution using Eq. (8).
 10. Phase 2 :
 11. Calculate new status of i th candidate solution based on phase 2 using Eq. (9).
 12. Update i th candidate solution using Eq. (10).
 13. end
 14. Save the best candidate solution so far.
 15. end
 16. Output the best obtained solution.
- End SOA.
-

3.3 Feature Extraction Using HLABP

An enhanced one-dimensional local feature extractor, HLABP stands for hexadecimal pattern. Combining signum and ternary functions [24] allows for feature extraction. Additionally, it makes use of two patterns that are variable: center-symmetric and linear. Figure 1 displays the patterns that were utilized.

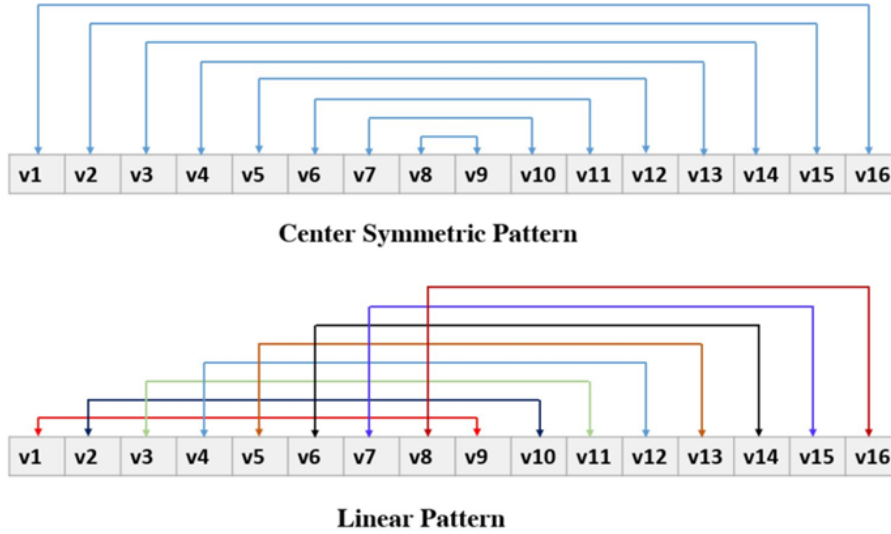


Figure 1. Used patterns

These patterns, illustrated in Figure 1, are used to implement the feature extraction method. We employed the signum and ternary binary feature extraction kernels. Figure 1 shows that eight relations are employed for feature extraction. Each pattern is extracted using eight bits from the signum function and sixteen bits from the ternary function. We used an 8-bit size to partition the extracted bits into non-overlapping units. Consequently, the suggested HLABP is used to extract 1536 characteristics. Signum and ternary functions are mathematically represented below.

$$\text{bit}^{\text{signum}}(i) = \text{signum}(x, y) = \begin{cases} 0, & x - y < 0 \\ 1, & x - y \geq 0 \end{cases} \quad (11)$$

$$\text{ter}(x, y) = \begin{cases} -1, & x - y < -thr \\ 0, & -thr \leq x - y \leq thr \\ 1, & x - y > thr \end{cases} \quad (12)$$

$$\text{bit}_{\text{lower}}^{\text{Ternary}}(i) = \begin{cases} 0, \text{ter}(x, y) > -1 \\ 1, \text{ter}(x, y) = -1 \end{cases} \quad (13)$$

$$\text{bit}_{\text{upper}}^{\text{Ternary}}(i) = \begin{cases} 0, \text{ter}(x, y) < 1 \\ 1, \text{ter}(x, y) = 1 \end{cases} \quad (14)$$

where, $\text{signum}(\cdot, \cdot)$ denotes the signum function, $\text{ter}(\cdot, \cdot)$ characterizes ternary purpose, $\text{bit}_{\text{lower}}^{\text{Ternary}}$ and $\text{bit}_{\text{upper}}^{\text{Ternary}}$ are the threshold values, and the lower and higher bits are taken from the ternary function. In signum and ternary functions, x and y are inputs. Using Eq. (15), the bits that were retrieved are transformed into decimal values.

$$\text{Decimal value} = \sum_{i=1}^8 \text{bit}(i) \times 2^{8-i} \quad (15)$$

3.4 Classification Using Deep Learning

This paper suggests a NCBAM to improve the performance of automated defect detection using sped-up signals. An attention mechanism is integrated into the proposed network utilizing CNNs. The attention module of NCBAM is fed by the CNN outputs. No changes are made to the input feature sizes by the NCBAM attention module. A convolution layer then applies the features' learned representation to the categorization outcomes. The spatial characteristics are extracted using a 33-layer residual network, which makes the network straightforward to train and optimize. The outstanding networks' design is exposed in Figure 2.

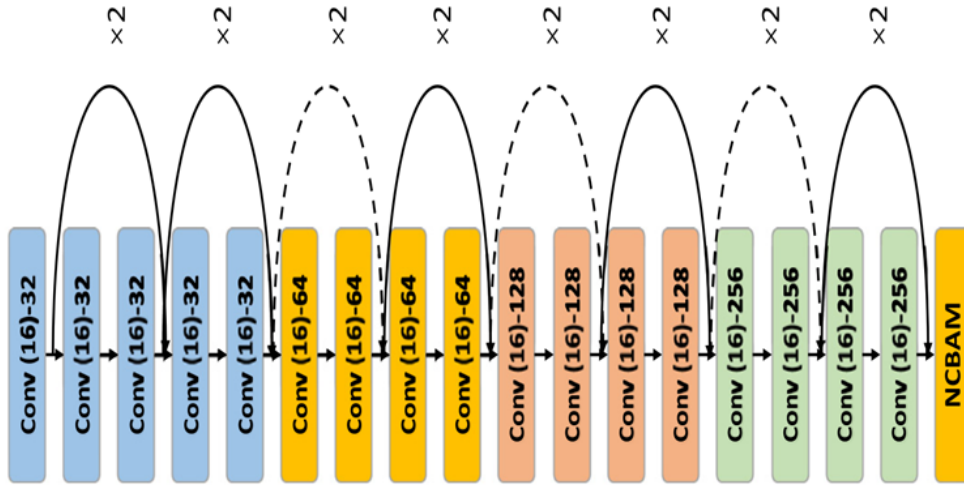


Figure 2. A 33-layer CNN perfect

Each set of filters (size=16) has a shortcut connection added to it. The 32-kernel, 16-size convolutional layer with a 1-step stride filters the input. A total of 64 16-kernel layers with a 1-step stride make up the second convolutional layer. The convolution kernel gets bigger as the convolution layer gets deeper. The residual module's dimensional irregularity is balanced using the 1×1 filter. Batch normalization and the rectified linear unit (ReLU) function follow each convolutional layer. The batch normalization layer speeds up deep neural network training and decreases the issue of internal covariate shift [25]. To avoid the vanishing gradient issue, one might use the ReLU activation function [26]. Figure 3 provides an overview of the NCBAM that has been proposed.

The form of the tensors represents the feature maps, such as $W \times C$ for C channels. “ \otimes ” means multiply by a matrix, and “ \oplus ” means add up each element individually. Row after row, the softmax operation is executed. 1D convolutions are shown by the blue boxes. In order to depict the connection between non-local blocks and CBAM, φ learns a weight matrix.

A spacetime non-local block plus a CBAM module make up the NCBAM. A CBAM module makes use of channel attention as well as spatial attention. Spatial attention is at the center of the informative input feature series. With channel attention, we may zero in on the input feature series' significant channel. To enhance the representational power of networks, a CBAM module applies pooling along the channel axis and input feature series [27]. A brief explanation of the CBAM is:

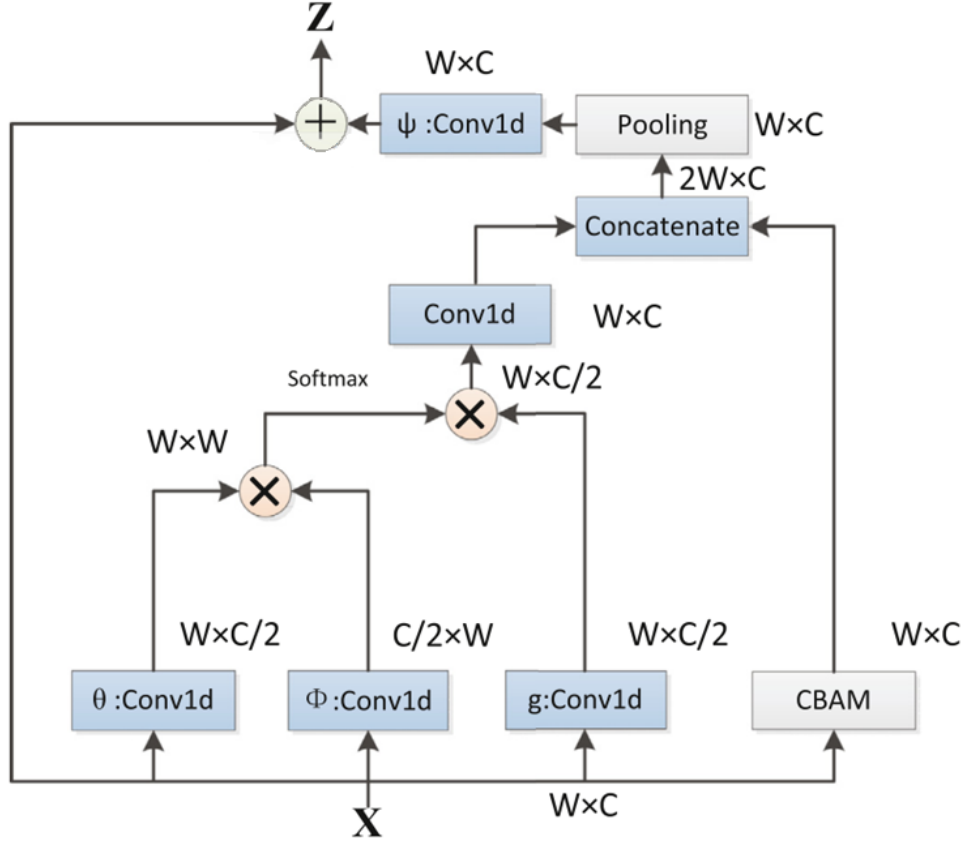


Figure 3. Design of the NCBAM

$$\begin{aligned} F' &= M_c(x) \otimes x \\ F'' &= M_S(F') \otimes F' \end{aligned} \quad (16)$$

where, $x \in R^{\text{channels} \times \text{length}}$ is an intermediate feature map. $M_c \in R^{\text{channels} \times 1}$ is a 1D channel attention map. $M_S \in R^{1 \times \text{length}}$ is a 1D spatial map. \otimes length is the input length following the convolution. After convolution, the number of channels is represented by channels. What comes out of CBAM is F'' . The non-local block differs from the CBAM in that it complements spatial and channel attention by concentrating on capturing long-range interdependence. A definition for the non-local block would be:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \quad (17)$$

The variable i is the index of the point in the intermediate feature map that needs to have its response computed, whereas the variable j represents all potential sites. x represents the in-between feature map that was input, and y is the block's output. The computation of a scalar between all j and i is done by the function f . Computed using a Gaussian function, similarity in an embedded space is the focus of this work. The representation of the input indication at position j is totalled by the function g . $C(x)$ is the factor of $\sum_{\forall j} f(x_i, x_j)$. In this paper, we define NCBAM as:

$$z = W(\text{pooling}(\text{cat}(y_i(x), F''(x)))) + x \quad (18)$$

where, $y_i(x)$ and $F''(x)$ are functions from Eqs. (16) and (17). The focus of the cat is on the spatial dimension. By combining resources, the concatenation may be made smaller. You may learn the crucial weight matrix, W , during training. The W stands for the connection between CBAM and a non-local block. One example of how this is put into practice is in the time series as 1D convolution. By drawing a connection between the traditional signal processing problem and CBAM, our study sheds light on the subject. A 1D layer, representing the unit of the sum of classes, is fed the features acquired from the NCBAM.

4 Results and Discussion

4.1 Implementation Data

The suggested representation's performance is tested with the help of the Python program. Two datasets, one from the Bearing Data Center and one from an automobile gearbox, were used to test the provided representation's ability to appropriately identify different types of faults [28]. The first dataset includes outer race bearing faults, small chipped gear faults, wasted tooth gear faults, and three types of compounds, resulting in seven different health statuses. One hundred examples with a half-second interval are created from 120,000,000 samples collected under each class name. Additionally, under each health condition, a set of 300 sample examples is obtained at different speeds. Finally, we have a dataset with 2100 test cases. You may find both datasets in the second dataset. In the second dataset, an inner, an outer race (OF), or a ball is one kind of bearing fault. As a result, ten distinct kinds of health conditions under different loads are present. Once again, a representation is created using the wavelet transform (WT) for each group of 2000 data points in each sample [29, 30]. Under each load, there are 60 instances of each health condition. To ensure that the method is effective, 2400 sample examples were collected.

4.2 Validation Analysis of Proposed Model

The existing models from the studies [18, 19, 22] are considered for validation analysis; however, they used different datasets for analysis. Hence, the existing models are implemented by using our datasets in the system, and the consequences are averaged in Table 1 and Table 2.

Table 1. Verification of projected model on gearbox dataset

Classifiers	Accuracy	Precision	Recall	F1-Score	AUC
SVM [19]	89.32	90.73	88.21	89.27	98.74
KNN [19]	89.46	90.21	88.21	89.31	98.43
CNN [18]	91.82	92.28	91.12	91.76	99.06
LSTM [18]	91.12	92.09	90.43	91.13	99.02
CNN-LSTM [18]	91.68	92.23	90.57	91.71	98.14
DBiRNN [22]	92.5	92.95	91.40	92.17	98.75
Proposed Model	94.17	94.28	93.76	93.93	99.43

Table 2. Analysis of proposed model on bearing dataset

Model	Accuracy	Recall	Precision	F1-Score	AUC
SVM [19]	86.90	86.14	87.47	83.60	92.14
KNN [19]	85.50	83.40	84.50	85.30	93.47
CNN [18]	88.50	87.40	88.10	88.30	94.10
LSTM [18]	92.00	91.90	91.65	91.99	94.69
CNN-LSTM [18]	89.57	90.70	89.66	89.65	95.79
DBiRNN [22]	93.00	92.91	92.45	92.65	96.47
Proposed Model	95.77	96.28	94.76	94.93	98.07

In the above Table 1, the representation indicates the verification of the proposed model on the gearbox dataset. In the analysis of the SVM [19] model, the accuracy was 89.32, the precision was 90.73, the recall was 88.21, the F1-score was 89.27, and the AUC range was 98.74, respectively. Similarly, the KNN [19] model achieved an accuracy of 89.46, precision of 90.21, recall of 88.21, F1-score of 89.31, and an AUC range of 98.43.

Moving on to the CNN [18] model, it attained an accuracy of 91.82, precision of 92.28, recall of 91.12, F1-score of 91.76, and an AUC range of 99.06. The LSTM [18] model achieved an accuracy of 91.12, precision of 92.09, F1-score of 90.43, recall of 91.13, and an AUC range of 99.02. The CNN-LSTM [18] model demonstrated an accuracy of 91.68, precision of 92.23, recall of 90.57, F1-score of 91.71, and an AUC range of 98.14.

For the DBiRNN [22] model, the accuracy was 92.5, the precision was 92.95, the recall was 91.40, the F1-score was 92.17, and the AUC range was 98.75. Finally, the proposed model achieved an accuracy of 94.17, precision of 94.28, recall of 93.76, F1-score of 93.93, and an AUC range of 99.43. These results are shown in Figure 4.

In Table 2, the analysis of the suggested model on the bearing dataset is described. In the SVM analysis [19], the model achieved an accuracy of 86.90, recall of 86.14, precision of 87.47, F1-score of 83.60, and an AUC range of 92.14, in that order. Subsequently, the KNN [19] model achieved an accuracy of 85.50, recall of 83.40, precision of 84.50, F1-score of 85.30, and an AUC range of 93.47, respectively.

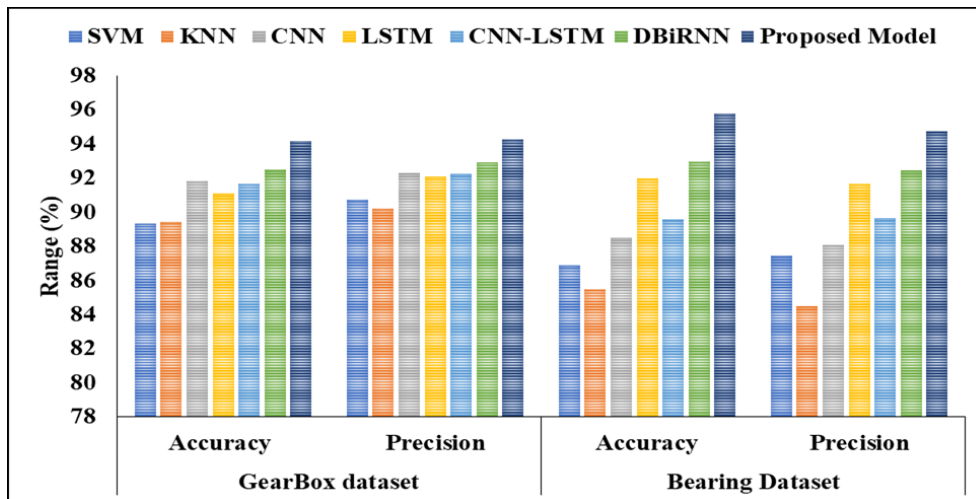


Figure 4. Graphical analysis of proposed model

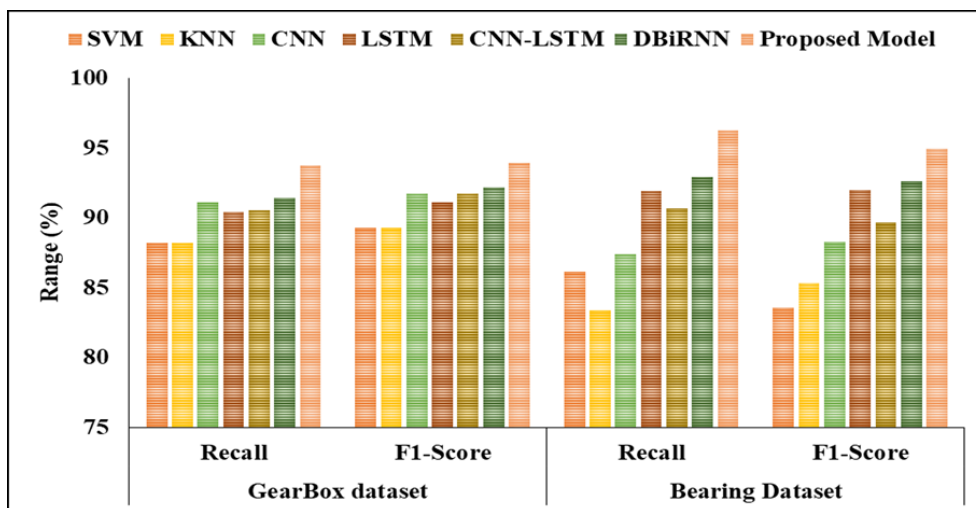


Figure 5. Visual representation of proposed model

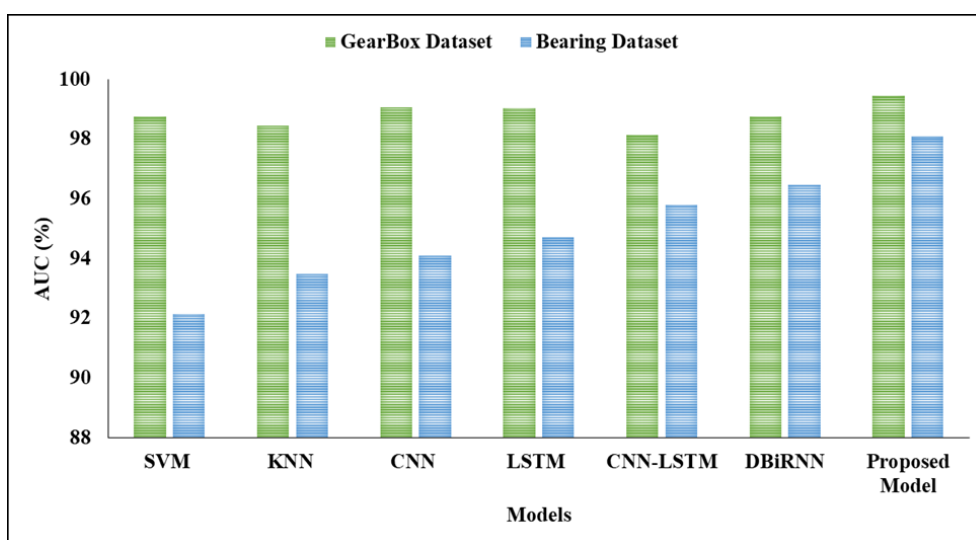


Figure 6. AUC verification

Moving on, the CNN [18] model achieved an accuracy of 88.50, recall of 87.40, precision of 88.10, F1-score of 88.30, and an AUC range of 94.10, respectively. Following that, the LSTM [18] model achieved an accuracy of 92.00, recall of 91.90, precision of 91.65, F1-score of 91.99, and an AUC range of 94.69.

Continuing, the CNN-LSTM [18] model achieved an accuracy of 89.57, recall of 90.70, precision of 89.66, F1-score of 89.65, and an AUC range of 95.79, respectively. Subsequently, the DBiRNN [22] model achieved an accuracy of 93.00, recall of 92.91, precision of 92.45, F1-score of 92.65, and an AUC range of 96.47.

Finally, the suggested model achieved an accuracy of 95.77, recall of 96.28, precision of 94.76, F1-score of 94.93, and an AUC range of 98.07, respectively. The comparative analysis is shown in Figure 5 and Figure 6.

5 Conclusion

This article introduces a model meant to detect faults in spinning machines. The data collecting procedure commences with the acquisition of information, which is then subjected to preprocessing and cropping of vibration signals utilizing the enhanced CWTS model. The CWTS approach utilizes the SOA to achieve optimization. To create the CNN model for fault classification, this research suggests the usage of a NCBAM. NCBAM incorporates spatio-temporal attention data to emphasize crucial qualities and conceal irrelevant ones. According to testing results utilizing the appropriate gearbox dataset (94.17% accuracy) and the bearing dataset (95.77% accuracy), the NCBAM model produced a superior average accuracy. The proposed model demonstrates its efficacy as a beneficial instrument for detecting issues with rotating machinery. The NCBAM model demonstrates potential for real-time defect diagnosis in numerous areas in the future. Integrating AI and machine learning algorithms can improve the precision and effectiveness of motor bearing issue diagnostics. These technologies have the capability to learn from vast datasets, boosting their ability to recognize subtle trends and predict possible problems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

- [1] R. Surendran, O. I. Khalaf, and C. A. Tavera Romero, "Deep learning based intelligent industrial fault diagnosis model," *Comput. Mater. Continua*, vol. 70, no. 3, 2022. <https://doi.org/10.32604/cmc.2022.021716>
- [2] R. M. Souza, E. G. Nascimento, U. A. Miranda, W. J. Silva, and H. A. Lepikson, "Deep learning for diagnosis and classification of faults in industrial rotating machinery," *Comput. Ind. Eng.*, vol. 153, p. 107060, 2021. <https://doi.org/10.1016/j.cie.2020.107060>
- [3] W. H. Li, R. Y. Huang, J. P. Li, Y. X. Liao, Z. Y. Chen, G. L. He, R. Q. Yan, and K. Gryllias, "A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges," *Mech. Syst. Signal Process.*, vol. 167, p. 108487, 2022. <https://doi.org/10.1016/j.ymsp.2021.108487>
- [4] Y. Xu, Z. Li, S. Wang, W. Li, T. Sarkodie-Gyan, and S. Feng, "A hybrid deep-learning model for fault diagnosis of rolling bearings," *Meas.*, vol. 169, p. 108502, 2021. <https://doi.org/10.1016/j.measurement.2020.108502>
- [5] J. Long, J. Mou, L. Zhang, S. Zhang, and C. Li, "Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots," *J. Manuf. Syst.*, vol. 61, pp. 736–745, 2021. <https://doi.org/10.1016/j.jmsy.2020.08.010>
- [6] K. Huang, S. Wu, F. Li, C. Yang, and W. Gui, "Fault diagnosis of hydraulic systems based on deep learning model with multirate data samples," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6789–6801, 2021.
- [7] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "A deep learning model for process fault prognosis," *Process Saf. Environ. Prot.*, vol. 154, pp. 467–479, 2021. <https://doi.org/10.1016/j.psep.2021.08.022>
- [8] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, and R. X. Gao, "WaveletKernelNet: An interpretable deep neural network for industrial intelligent diagnosis," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 52, no. 4, pp. 2302–2312, 2021.
- [9] X. Yu, Z. Zhao, X. Zhang, Q. Zhang, Y. Liu, C. Sun, and X. Chen, "Deep-learning-based open set fault diagnosis by extreme value theory," *IEEE Trans. Ind. Inform.*, vol. 18, no. 1, pp. 185–196, 2021.
- [10] M. Q. Tran, M. Elsis, K. Mahmoud, M. K. Liu, M. Lehtonen, and M. M. Darwish, "Experimental setup for online fault diagnosis of induction machines via promising IoT and machine learning: Towards industry 4.0 empowerment," *IEEE Access*, vol. 9, pp. 115 429–115 441, 2021. <https://doi.org/10.1109/ACCESS.2021.3105297>

- [11] M. Xia, H. Shao, D. Williams, S. Lu, L. Shu, and C. W. de Silva, "Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning," *Reliab. Eng. Syst. Saf.*, vol. 215, p. 107938, 2021. <https://doi.org/10.1016/j.res.2021.107938>
- [12] T. Han, C. Liu, R. Wu, and D. Jiang, "Deep transfer learning with limited data for machinery fault diagnosis," *Appl. Soft Comput.*, vol. 103, p. 107150, 2021. <https://doi.org/10.1016/j.asoc.2021.107150>
- [13] J. Yu, C. Zhang, and S. Wang, "Multichannel one-dimensional convolutional neural network-based feature learning for fault diagnosis of industrial processes," *Neural Comput. Appl.*, vol. 33, pp. 3085–3104, 2021. <https://doi.org/10.1007/s00521-020-05171-4>
- [14] A. Khorram, M. Khaloeei, and M. Rezaghi, "End-to-end CNN+ LSTM deep learning approach for bearing fault diagnosis," *Appl. Intell.*, vol. 51, pp. 736–751, 2021. <https://doi.org/10.1007/s10489-020-01859-1>
- [15] S. Baswaraju, V. U. Maheswari, K. K. Chennam, A. Thirumalraj, M. P. Kantipudi, and R. Aluvalu, "Future food production prediction using AROA based hybrid deep learning model in agri-sector," *Hum.-Centric Intell. Syst.*, pp. 1–16, 2023. <https://doi.org/10.1007/s44230-023-00046-y>
- [16] M. Zabin, H. J. Choi, and J. Uddin, "Hybrid deep transfer learning architecture for industrial fault diagnosis using hilbert transform and DCNN–LSTM," *J. Supercomput.*, vol. 79, no. 5, pp. 5181–5200, 2023. <https://doi.org/10.1007/s11227-022-04830-8>
- [17] B. Li and C. Zhao, "Federated zero-shot industrial fault diagnosis with cloud-shared semantic knowledge base," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11 619–11 630, 2023. <https://doi.org/10.1109/JIOT.2023.3243401>
- [18] J. Shin and S. Lee, "Robust and lightweight deep learning model for industrial fault diagnosis in low-quality and noisy data," *Electronics*, vol. 12, no. 2, p. 409, 2023. <https://doi.org/10.3390/electronics12020409>
- [19] E. Brusa, L. Cibrario, C. Delprete, and L. G. Di Maggio, "Explainable AI for machine fault diagnosis: Understanding features' contribution in machine learning models for industrial condition monitoring," *Appl. Sci.*, vol. 13, no. 4, p. 2038, 2023. <https://doi.org/10.3390/app13042038>
- [20] Y. Chen, Z. Chen, S. Guo, Y. Zhao, Z. Liu, P. Wu, and H. Yu, "Efficient training of large-scale industrial fault diagnostic models through federated opportunistic block dropout," *arXiv preprint*, 2023. <https://doi.org/10.48550/arXiv.2302.11485>
- [21] C. Chen, C. Liu, T. Wang, A. Zhang, W. Wu, and L. Cheng, "Compound fault diagnosis for industrial robots based on dual-transformer networks," *J. Manuf. Syst.*, vol. 66, pp. 163–178, 2023. <https://doi.org/10.1016/j.jm.2022.12.006>
- [22] M. M. Alnfai, "Improved symbiotic organism search with deep learning for industrial fault diagnosis," *Comput. Mater. Continua*, vol. 74, no. 2, 2023. <https://doi.org/10.32604/cmc.2023.033448>
- [23] S. Guo, T. Yang, W. Gao, and C. Zhang, "A novel fault diagnosis method for rotating machinery based on a convolutional neural network," *Sensors*, vol. 18, no. 5, pp. 1–16, 2018. <https://doi.org/10.3390/s18051429>
- [24] T. Tuncer and S. Dogan, "Pyramid and multi kernel based local binary pattern for texture recognition," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, pp. 1241–1252, 2020. <https://doi.org/10.1007/s12652-019-01306-1>
- [25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. <https://doi.org/10.48550/arXiv.1502.03167>
- [26] S. S. Talathi and A. Vartak, "Improving performance of recurrent neural network with relu nonlinearity," *arXiv preprint arXiv:1511.03771*, 2015. <https://doi.org/10.48550/arXiv.1511.03771>
- [27] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," *arXiv preprint arXiv:1807.06521*, 2018. <https://doi.org/10.48550/arXiv.1807.06521>
- [28] R. Saravanakumar, N. Krishnaraj, S. Venkatraman, B. Sivakumar, and S. Prasanna, "Hierarchical symbolic analysis and particle swarm optimization based fault diagnosis model for rotating machineries with deep neural networks," *Measurement*, vol. 171, no. 108771, pp. 1–8, 2021. <https://doi.org/10.1016/j.measurement.2020.108771>
- [29] N. Sulaiman, G. Abdulsahib, O. Khalaf, and M. N. Mohammed, "Effect of using different propagations of OLSR and DSDV routing protocols," in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation, Langkawi, Malaysia*, 2014, pp. 540–545. <https://doi.org/10.1109/ISMS.2014.99>
- [30] S. Sengan, P. V. Sagar, R. Ramesh, O. I. Khalaf, and R. Dhanapal, "The optimization of reconfigured real-time datasets for improving classification performance of machine learning algorithms," *Math. Eng. Sci. Aerosp.*, vol. 12, no. 1, pp. 1–21, 2021.