SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I1P112

Volume 12 Issue 1, 129-141, January 2025
© 2025 Seventh Sense Research Group®

Original Article

Performance Analysis of Neural Networks for Fault Detection in Induction Motor

N. Sivaraj[1], B. Rajagopal[2]

[1,2]*Department of Electrical and Electronics Engineering, Annamalai University, Tamilnadu, India.*

[1]*Corresponding Author : nsivaraj.phd@gmail .com*

Received: 09 November 2024 Revised: 11 December 2024 Accepted: 12 January 2025 Published: 25 January 2025

Abstract - *Condition monitoring and fault identification are essential for preventing damage in industrial machinery, particularly in three-phase squirrel cage Induction Motors (IMs), which are widely used due to their reliability and robust design. This paper compares three advanced techniques for diagnosing Motor faults, including issues like inter-turn faults in the stator, bearing malfunctions and faults from the rotor by analyzing motor current and speed: 1-D Convolutional Neural Networks (1-D CNN), Grey Wolf Optimized Probabilistic Neural Networks (GWOPNN), and Whale Optimized Pattern Recognition Neural Networks (WOAPRNN). The study evaluates each method's ability to detect and classify faults. Results show that the whale-optimized pattern classification neural network achieves the highest accuracy of 99.15%, making it the most effective method for fault detection. Each technique -offers unique strengths in fault classification and detection, with the goal of enhancing motor reliability and efficiency in industrial environments. By improving fault diagnosis, these methods contribute to reducing downtime, lowering maintenance costs, and increasing the operational lifespan of induction motors.*

Keywords - *1-D CNN, Bearing faults, Fault diagnosis, GWOPNN, Induction motors, Machine learning, Rotor faults, Stator Interturn faults, WOAPRNN.*

1. Introduction

In recent years, electrical machineries has been widely utilized in the manufacturing industry due to its versatility, efficiency, and ability to handle a variety of industrial tasks. Induction motors, like other machines, are important for powering the equipment, driving the machineries and boosting overall productivity. With the rapid development and improvement of technology and science, electrical machines function on a regular basis practically for all applications. These machines often work in tough conditions like humidity, dust, and heavy loads. Without regular maintenance, they can cause money loss, lower production, and safety problems. In industries, electrical machines play a vital role in powering processes, running various equipment, and enabling automation. These rotating machines are made up of static components like the stator and rotating components like the rotor, shaft, and bearings [1–5]. For commercial and industrial applications, the IM is normally employed due to their low cost and ruggedness [6]. However, in certain operating conditions, an induction motor can suddenly malfunction due to lack of attention and inadequate Sustenance. Faults cause motors to fail [7], with examples including rotor faults, bearing faults [8, 9], and stator interturn faults [10]. These faults can significantly affect motor performance, leading to inefficiencies, overheating, and eventual failure if not addressed promptly. Early detection and diagnosis are essential for maintaining motor reliability and preventing operational disruptions in industrial environments. In [11], Genetic Algorithm (GA) for feature extraction along with classification algorithms, including random forest and decision tree, were evaluated to analyze bearing faults. According to their experimental results, the proposed scheme achieved high accuracy in distinguishing and recognizing the faults in different induction motor conditions. However, it was found to be ineffective for real-time predictions, limiting its practical application in dynamic industrial environments. Vibration analysis and MCSA are two Particularly notable generally utilized fault detection methods in induction motors [12, 13]. Another method for fault diagnosis relies on spectrum analysis using the Fourier Transform (FT) to identify various faults by examining the frequency spectrum, which displays different harmonics, including fundamental and adjacent fault frequencies. These harmonics vary between healthy and faulty motors, creating distinctive signatures for mechanical and electrical faults. However, the implementation of these analyses can be relatively expensive. Alternatively, some methods utilize acoustic signal processing to analyze rotor noise as a means of detecting faults [14]. The objective is to recognize and derive signal factors related to faults. The Fourier Transform (FT) has been deployed to analyze noise for this purpose [15]. By examining the frequency components of the noise signals, the FT helps


This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

identify characteristics that are indicative of specific faults, facilitating more accurate fault diagnosis and maintenance strategies. From the literature survey, it was evident that machine learning algorithms, when combined with proper optimization techniques, provide better results compared to other methods. In this research, three neural networks were trained: 1-D CNN, Whale Optimized Neural Network, and Grey Wolf Optimized Probabilistic Neural Network. These networks were trained to detect faults in the stator, rotor, and bearings, as well as identify healthy conditions. Their performance in accurately detecting faults was tested and evaluated. Also, many researchers have primarily considered only stator current for analysis in their fault diagnosis studies. However, in this work, both stator current and speed were considered, which is a novelty. This approach leads to more accurate fault predictions, making the method better at identifying the type of fault. By incorporating both stator current and speed, the model captures a wider range of motor characteristics, improving the reliability and accuracy in fault diagnosis.

2. Proposed System Description

Figure 1 illustrates a block diagram of the method proposed. The analog speed and current signals from the induction motor are first collected and converted into digital form during preprocessing. This makes it easier to process the signals. Digital filters like the Wiener filter, Hilbert Transform, and Modified Gabor filter are then used to process the data. After applying these filters, three separate sets of features are extracted. These features include the minimum, maximum, standard deviation, mean, and median, which are important for training neural networks. Features from the Wiener filter are used to train a 1D-Convolutional Neural Network (1D-CNN). The Hilbert Transform was utilized to extract relevant features for training the WOPRNN, and the Modified Gabor filter was used for training the GWOPNN. These three neural network classifiers are trained independently to identify different fault types: bearing faults, broken rotor faults, and inter-turn faults. Each classifier is tested, and the results are discussed in detail in the results and discussion section.

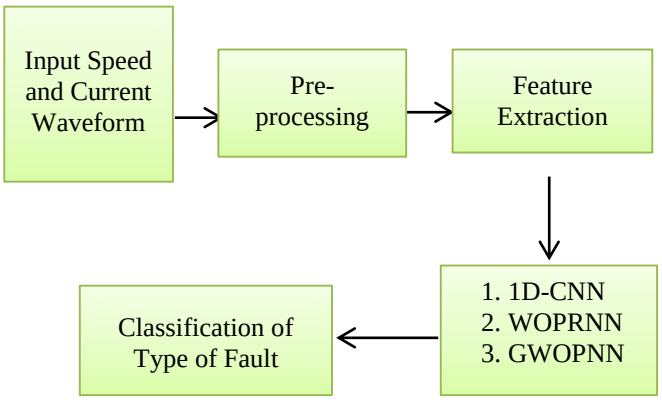


Fig. 1 Proposed system block diagram

2.1. GWO Algorithm

The Grey Wolf Optimizer (GWO) algorithm draws its inspiration from the cooperative hunting and social hierarchy techniques exhibited by grey wolves. The population of grey wolves was optimized mathematically by resampling the surrounding, attacking, tracking and hunting processes. The hunt for grey wolves consists of 3 steps: prey encircling, social hierarchy stratification and attacking hunter [16-17].

2.1.1. Fitness Evaluation in GWO

Grey Wolf Optimization (GWO) works by updating the positions of wolves (candidate solutions) based on their best fitness values. Fitness functions depend on the specific problem, but in general, for a classification problem, it could be accuracy, error rate, or other objective measures like Mean Squared Error (MSE).

$$Fitness = \frac{\mathrm{TP+TN}}{TN+FP+TP+FN} \quad (1)$$

Where: FP False positives, TN True negatives, TP True positives, FN False negatives

2.1.2. Updating Grey Wolves Position

The position of the wolves in the GWO algorithm was updated by alpha, beta, and delta.

$$\vec{X}(t+1) = \frac{\vec{X}_\alpha(t)+\vec{X}_\beta(t)+\vec{X}_\delta(t)}{3} \quad (2)$$

Where $\vec{X}(t)$ is the grey wolf's position at time t, and α, β, and δ are the three best wolf's positions.

2.1.3. Component Update Rules

$$\vec{A} = 2.\vec{a}.\vec{r} - \vec{a} \quad (3)$$

$$\vec{C} = 2.\vec{r} \quad (4)$$

Where, $\vec{a}$ is a coefficient vector that decreases linearly from 0 to 2 over iterations. $\vec{r}$ is a random vector in [0, 1]. The new position of the grey wolf is calculated by:

$$\vec{X}(t+1) = \overrightarrow{X_a}(t) - \vec{A}.\left|\vec{C}.\overrightarrow{X_a}(t) - \vec{X}(t)\right| \quad (5)$$

Similar expressions are used for $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$. The network chooses the class with the greatest likelihood based on posterior probability. The current optimization mechanism is iteratively applied across all agents, ensuring the solution converges towards the optimal position in the search space.

2.2. Implementation of GWO with PNN

In a PNN, there are typically two hidden layers and one input layer. The pattern units are located in the first hidden layer. Contrary to the Back propagation neural network approach, the PNN approach has a significant advantage in

that it requires only one learning step. Backpropagation NN learning can be akin to learning through trial and error, whereas the PNN gains knowledge from experience rather than relying solely on trial and error. The PNN boasts a robust framework and highly effective operations. It can perform effectively even with a limited amount of training instances. The weights w(ij), which are determined using the PNN approach as illustrated in Figure 2, are multiplied by the output of the input dataset and then transferred to the pattern layer. A transfer function is applied to transform these weights into the summation and output layers, as previously demonstrated. The output layer typically consists of only one class because a single output is usually required. During the training process, the primary goal is to identify the most accurate weights assigned to connector lines. To achieve this, the GWO technique was selected to obtain optimized PNN training parameter settings, ultimately leading to a higher level of accuracy.

This combination of PNN and GWO offers a powerful tool for pattern recognition and classification tasks. Figure 3 illustrates the flowchart of the GWOPNN process. Input features derived from current and speed data are provided as inputs to the PNN. In this research, for training, 80% of the dataset was designated, and the remaining 20% was set aside for testing. Control parameters were configured for the GWOPNN. During the training of the PNN with the GWO algorithm, its fitness function was calculated. Then, the position of the GWOPNN was updated iteratively using the GWO algorithm. This technique helps select and find the best control parameters to optimize the PNN. After that, the dataset is used to train the PNN with the chosen control parameters to build the best model for the task. Throughout the optimization procedure, the wolves' positions were adjusted based on Equations (2), (3), (4), and (5), while their fitness values were computed using Equation (1). This optimization process continues until the best possible solution is found, ensuring optimal performance of the PNN. This repeated procedure enables the model to adjust based on the fitness values, resulting in enhanced fault detection accuracy. As this algorithm converges, the PNN becomes more efficient in predicting faults.

2.2.1. Modified Gabor Filter (MGF)

Figure 4 illustrates the working of the MGF, where a sinusoidal carrier wave and Gaussian envelope are combined to analyze and extract information from the signals [18]. This is a tunable filter capable of highlighting features of a specific input signal. By applying this filter to the input signal, the similarity between the input signal and the Gabor filter at each point of the signal can be assessed. The features and patterns of the input signal are evaluated based on the characteristics defined by the Gabor filter at each point of the signal. Equation (6) expresses a modified Gabor filter.

$$MGF = e^{\left(\frac{-(t-\mu)^2}{2*\sigma^2}\right)} * \cos(2 * \pi * f * t) \quad (6)$$

Where 't' is the time domain variable, 'μ' determines the centre position of the MGF in the time domain. It represents the mean or central position of the filter. It controls the spread or width of the Gaussian envelope. 'f' is the frequency parameter that controls the spatial frequency of the sinusoidal part of the filter.

$$G(t) = \int_{-\infty}^{\infty} s(\tau) * MGF(t-\tau)d\tau \quad (7)$$

In Equation (7), G (t) is the Gabor filtered signal; s (τ) is the signal that we want to filter using the modified Gabor filter;'*' This function is used to perform convolution between the 's(τ)' and the 'modified gabor filter'. Convolution is a mathematical operation that combines the filter with the input signal to produce an output signal.

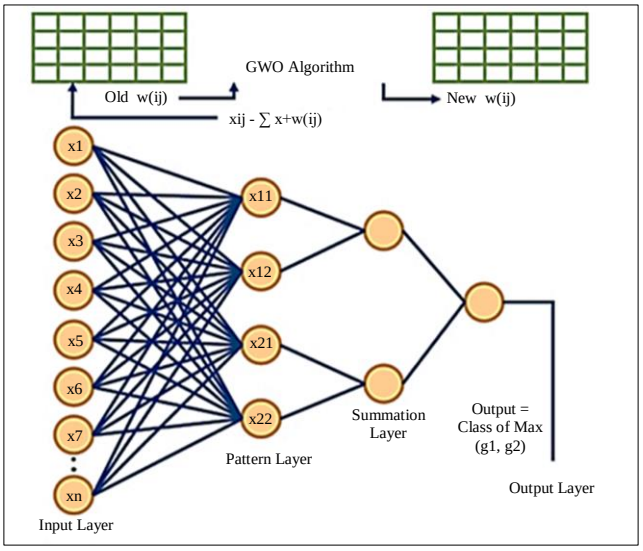


Fig. 2 Mechanism of GWO algorithm with PNN

2.3. Whale Optimized Algorithm (WOA)

WOA, like other metaheuristic optimization algorithms, begins by producing a population of random candidate remedies from which the global optimum solution to the problem is discovered. Based on its structure, this algorithm continuously enhances and modifies the solution till the optimal value is obtained. The WOA rules' ability to adapt and develop the outcome is the main way it differs from other metaheuristic algorithms. The instinct of a whale to follow prey, which is accomplished by circling around the prey, constructing a trap, and then attacking it, has an impact on the WOA. This is known as bubble-net feeding behavior. Before attacking its prey, the humpback whale creates bubbles by spiralling around it. This feeding behavior inspires the WOA's main structure.

2.3.1. Prey Encircling

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}.\overrightarrow{D} \; if \; p < 0.5 \\ \overrightarrow{D'}.e^{bl}\cos(2\pi l) + \overrightarrow{X^*}(t) if \geq 0.5 \end{cases} \quad (8)$$

$$\vec{D} = \left|\vec{C}.\vec{X^*}(t) - \vec{X}(t)\right| \quad (9)$$

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \quad (10)$$

$$\vec{C} = 2.\vec{r} \quad (11)$$

Where t denotes the current iteration and r denotes random constants in the range [1, 1]. The logarithmic spiral shape denoted by b and a decreases linearly from 2 to 0 over the iteration.

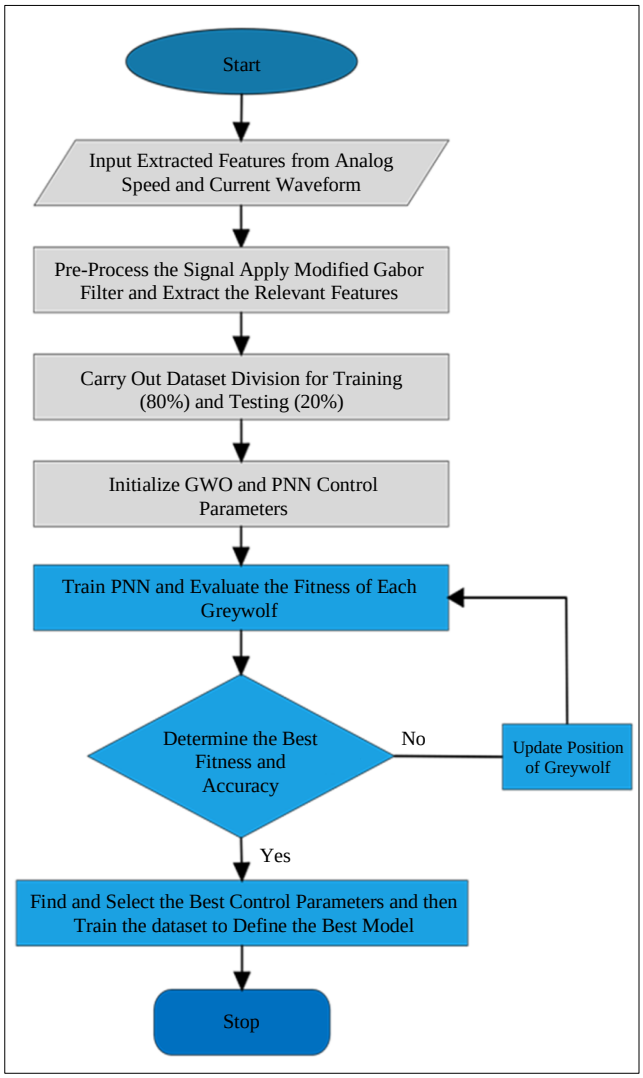


Fig. 3 Flowchart of GWOPNN

2.3.2. Spiral Updating Position

This behavior of whales is mathematically modeled as,

$$\vec{X}(t+1) = \vec{D'}.e^{bl}\cos(2\pi l) + \vec{X^*}(t) \quad (12)$$

Where, $\vec{D'}$ indicates the distance separating the prey from the whale, b is used to set the shape of the logarithmic spiral, and l is a random value drawn from the interval [-1, 1]].

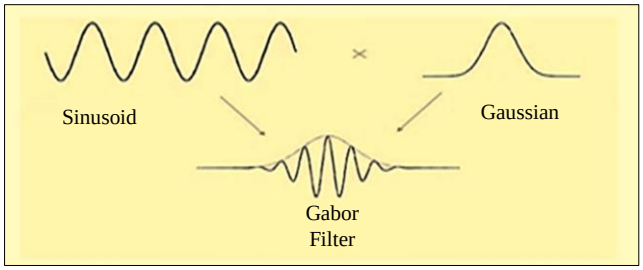


Fig. 4 Modified gabor filter

2.3.3. Implementation of WOA with PRNN

In a PRNN, the output of the hidden layers and the connections between different layers are determined by the weights and biases. The objective of training this NN is to adapt these weights and biases in order to reduce the disparity between the networks, the actual target values and the predicted output.

In Equation (13), Mean Squared Error (MSE) is calculated as follows:

$$\text{MSE} = \frac{-\sum_{i=1}^{n}(O_i - \widehat{O_i})^2}{n} \quad (13)$$

Where o_i represents the actual output of a given input sample i.,$\widehat{o_i}$ represents the expected output (target) of a given input sample i.

$$U' = \frac{U - U_{min}}{U_{max} - U_{min}} \quad (14)$$

To validate the model's accuracy, the optimal biases and weights obtained during the training phase parameters are applied during the testing phase. The WOAPRNN's exceptional capability to overcome the local optima problem and determine the optimal weights and biases for the neural network enhances the accuracy and effectiveness of the proposed model. Figure 6 presents a flowchart detailing the application of the Whale Optimization Algorithm (WOA) for optimizing a neural network designed for pattern recognition. It starts by randomly initializing a population of whale positions, which represent candidate neural network solutions. The fitness of each whale is evaluated using the Mean Squared Error (MSE). Based on the probability *p*, each whale's position is updated either through prey encircling Equations (8) and (9) or spiral updating Equation (12). The coefficient A for controlling the position update is calculated using Equation (10). If a superior solution is found in an iteration, the optimal solution X∗(t) is adjusted. If the iteration reaches its maximum, this process ends, and the whale with the lowest MSE is returned as the best neural network configuration.

2.3.4. Hilbert Transform (HT)

HT helps analyze a signal's instantaneous amplitude, frequency, and phase by converting a real-valued signal into its analytic form. This makes it a valuable tool for tasks such

as modulation, demodulation, and fault diagnosis. The HT is mathematically expressed by:

$$X(t) = x(t) + jh\{x(t)\} \quad (15)$$

Here, X (t) is the complex analytical signal, x (t) is the real-valued input signal, and j represents the imaginary unit. This indicates that the positive frequency components are preserved, whereas the negative frequency components are removed in the frequency domain representation of the analytic signal.

2.4. 1D-CNN

1D-CNNs are a specific type of Convolutional Neural Network (CNNs) designed for processing one-dimensional data. While traditional CNNs are predominantly used for image data, 1D-CNNs are tailored for tasks involving sequences, time series, audio, and other one-dimensional data. They excel in capturing local patterns and dependencies within such data.

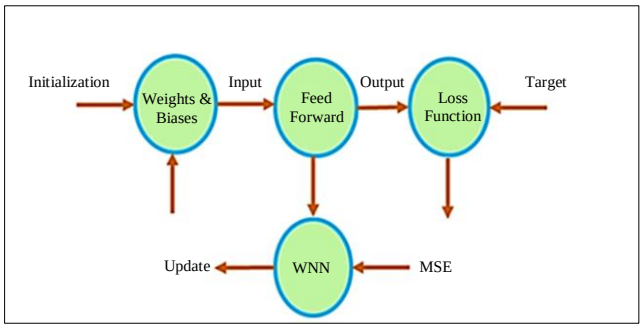


Fig. 5 Whale-optimized PRNN

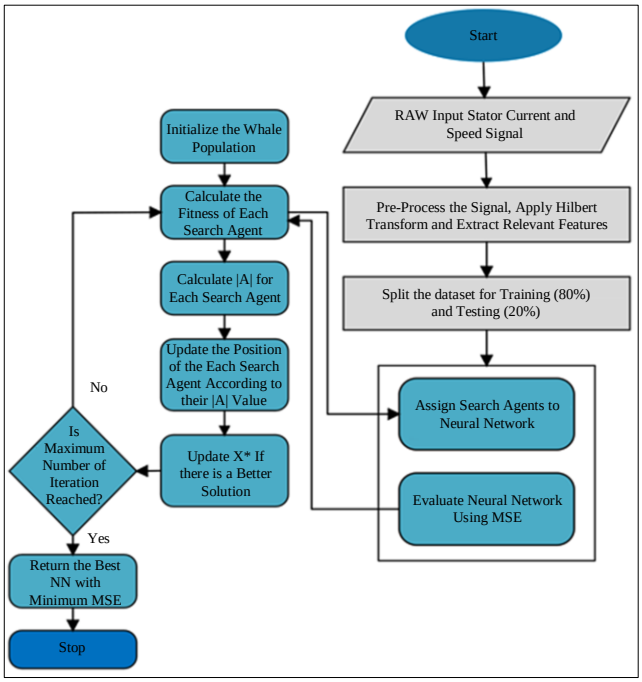


Fig. 6 WOA implementation algorithm

2.4.1. 1D Convolution Operation

At the core of 1D-CNNs is the 1D convolution operation. Given an input sequence or signal' x' of length 'N' and a filter (also called a kernel) 'ω' of length 'L,' the convolution operation of the feature map is generated by moving the filter over the input. The output at each position' j' is computed as follows:

$$c(j) = f(\sum_{i=0}^{L-1} \omega(i)x(j-i) + b), j = 0,,N-1 \quad (16)$$

Here, 'c (j)' is the output at position 'j'; 'f (·)' is the activation function, often ReLU or another non-linear function; 'ω (i)' represents the filter weights; 'x (j - i)' denotes the input value at the corresponding position; 'b' is the bias term. The convolution operation produces a feature map 'c' that captures local patterns in the input sequence.

2.4.2. Pooling Layers(PL)

In PL feature maps, spatial dimensions were reduced. Max pooling, for example, chooses the maximum value within a m X 1 local window, moving with a stride 's.' This helps retain essential information while reducing computational complexity.

$$d=\max(u(m\times 1, s)c) \quad (17)$$

Here, 'd' is PL output; 'u' is the max-pooling operation; 'm × 1' specifies the size of the pooling window; 's' is the stride.

2.4.3. Fully Connected Layer and Output

The feature map generated after pooling and convolution is flattened and sent through one or more Fully Connected (FC) layers. These layers establish connections among all neurons in the preceding layer, enabling intricate combinations of features. The output from the FC layer is typically passed through an activation function (such as Softmax for classification) to yield the final predictions.

2.4.4. Loss Function

A loss function is utilized to train the network, measuring the disparity between the predicted output and the actual values.

$$l = \sum_{k=1}^{c} t(k)\log(p(k)) \quad (18)$$

Here: 'l' denotes loss; 'C' denotes the total number of classes; 't(k)' refers to the actual label. The predicted probability of the input belonging to class' k' is denoted as 'p(k)'.

2.4.5. Training

Training is typically carried out using backpropagation and optimization algorithms like Adam. The gradients calculated during the backpropagation process are employed

to adjust the network's biases and weights, thereby reducing the loss function. Figure 7 illustrates the architecture of 1D-CNN.

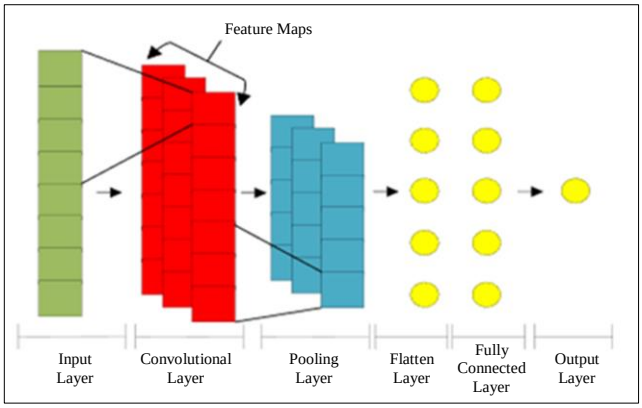


Fig. 7 Architecture of 1D CNN

2.5. Wiener Filter

The Wiener filter is a filtering technique based on linear principles intended to reduce the MSE between the target signal and its approximation, effectively reducing noise. It is commonly utilized in signal processing for various purposes, including noise reduction, image restoration, and system identification.

2.5.1. Noisy Signal Model

The observed signal z(t) (whether current or speed) consists of the true signal x(t) combined with some noise y(t):

$$z(t) = x(t) + y(t) \tag{19}$$

Where z(t) is the noisy signal (current or speed), x(t) is the true signal we want to recover, and y(t) is the unwanted noise.

2.5.2. Wiener Filter Formula

The Wiener filter works in the frequency domain to reduce noise. The filter H(f) at any frequency f is calculated as:

$$\mathrm{H(f)} = \frac{\mathrm{S_{xx}(f)}}{\mathrm{S_{xx}(f)+S_{nn}(f)}} \tag{20}$$

Equation (20) adjusts the filter to allow frequencies with more signal power and suppress those dominated by noise.

2.5.3. Filtered Signal

To get a clean signal, we apply the Wiener filter to the noisy signal's frequency components:

$$\hat{x}(f) = y(f).H(f) \tag{21}$$

Where, $\hat{x}(f)$ the filtered signal, $y(f)$ the noisy signal.

2.5.4. Convert Back to Time Domain

After filtering, the filtered signal is converted back to the time domain:

$$\hat{x}(t) = F^{-1}|\hat{x}(f)| \tag{22}$$

Where, F^{-1} is the inverse Fourier transform, which gives us the clean signal in time form $\hat{x}(t)$.

2.5.5. Minimizing Error

The filter is designed to minimize the overall error by reducing the difference between the true signal x(t) and the estimated signal. $\hat{x}(t)$. By iteratively adjusting the filter parameters, the algorithm ensures obtaining a closer original signal.

3. Generating Synthetic Fault Data

In addition to the healthy conditions, the neural network training also includes data from faulted scenarios (broken rotor faults, bearing faults, and interturn faults) under different loading conditions. This ensures that the model learns how faults impact the motor's performance across various load conditions. By including both healthy and faulted data across different loads, the neural network can better detect and classify faults based on rotor speed and current. Figure 8 shows synthetic fault data generated by deliberately damaging a 6205-2Z bearing, introducing faults in the outer ring, inner race, and ball bearings. The bearing dimensions are twenty-five mm bore diameter, outer diameter, fifty-two mm and fifteen mm width. Faulty bearings were successively installed in a 3-phase induction motor, and speed and current waveforms were recorded under various loads. These patterns were combined with modeling motor behavior during bearing faults, and these data helped in training the neural network.

The trained network was then tested on both synthetic and real-world fault data to evaluate its performance in accurately classifying different types of bearing faults. To generate synthetic faults in the rotor bars, holes were drilled into a rotor with 32 bars. Each hole was 10 mm deep and 8 mm in diameter, breaking the electrical continuity.

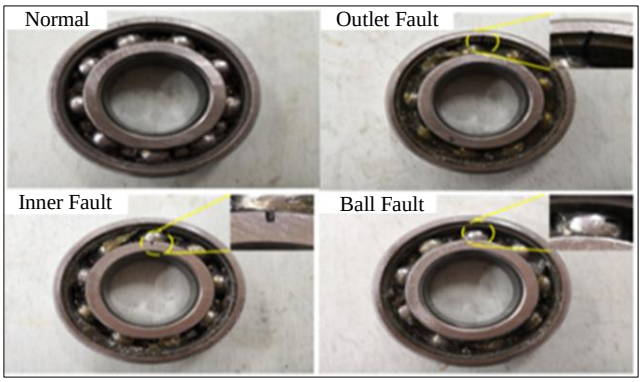
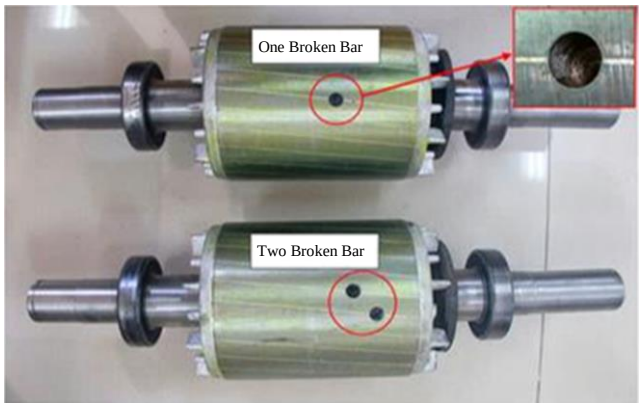


Fig. 8 Simulated faults on healthy, outer ring, inner ring and ball in bearing

(a)



(b)

Fig. 9 Rotor shaft: (a) Healthy bar, and (b) Single bar and double bars broken.

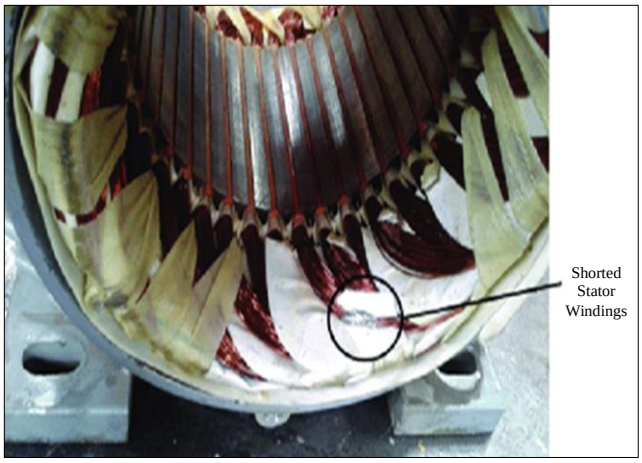


Fig. 10 Shorted stator windings

Figure 9 (a) and (b) show images of both healthy and damaged rotor bars. Figure 8 simulated faults on the healthy, outer ring, inner ring and ball in the bearing. Initially, a healthy rotor bar was installed in a 3-phase induction motor, and the current and speed characteristics were recorded. Then, rotors with one and two broken bars were tested similarly. Data for stator current and speed characteristics were collected under all three load conditions. Each motor setup was run 100 times, with the corresponding speed and current characteristics recorded, replicating the motor's behaviour under rotor bar faults. Figure 10 shows short stator windings. An induction motor with 36 stator slots and coils consisting of 300 turns each was used. To create interturn faults, insulation was removed from the windings, and tapings were removed at intervals of 10, 20, and 30 turns in phase A. These modified windings were then shorted individually. The motor was run 100 times under no load, full load, and half load conditions to collect speed and current data. This data, mimicking motor behavior during inter-turn faults, was utilized for training the neural network model for fault diagnosis.

4. Experimental Setup of Proposed System

The experimental setup, shown in Figure 11, was designed to study the performance of a 1.5 kW three-phase induction motor under both healthy and faulted conditions. The process of synthetic fault generation, as detailed in the previous section, was implemented to simulate various fault conditions.

This included introducing a damaged bearing for bearing faults, breaking a rotor bar for rotor faults, and short-circuiting the stator winding to simulate stator faults. Figure 11 demonstrates the experimental layout of the proposed system, which comprises the NI MyDAQ for data acquisition from the 3-phase induction motor. The NI MyDAQ is compatible with LabVIEW. Various states of healthy and fault characteristics of the current and speed of the induction motor were obtained using the LA55P current transducer and the Hall Effect sensor, respectively. To capture the speed characteristics, neodymium magnets were strategically placed on the shaft of the 3-phase induction motor. The Hall Effect sensor output is interfaced with the digital pins of the NI MyDAQ, and a Virtual Instrument (VI) in LabVIEW is used to calculate pulse signals generated by the magnetic effect on the sensor.

Fig. 11 Experimental setup of the proposed system

Figure 12 shows how pulse signals were obtained. When the magnets were facing north, the output voltage was high, and when the magnets were facing south, the output voltage was low. After calculating the pulse length, the speed of the induction motor was determined. Similarly, another VI was set up to trace the current characteristics using the LA55P transducer, with its outputs connected to the analog inputs of the MyDAQ. This helps in extracting the current characteristics of the induction motor under specific fault and loading conditions.

This waveform is then used in the subsequent section for feature extraction. Additionally, a Mixed-Signal Oscilloscope (MSO) is employed to verify the 3-phase current waveforms from the stator. Table 1 provides the specifications of the 1.5 kW 3-phase induction motor used in this experimental setup. The laptop used for this experiment is equipped with an Intel i7 processor, a RAM capacity sixteen GB, and runs on the Windows 10 OS. The data collected from these instruments is pre-processed using noise reduction techniques before feature extraction.

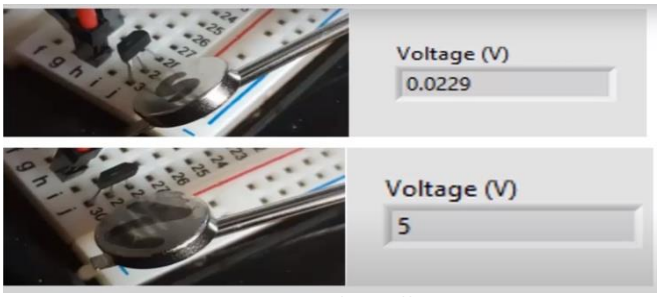


Fig. 12 Placement of hall effect sensor

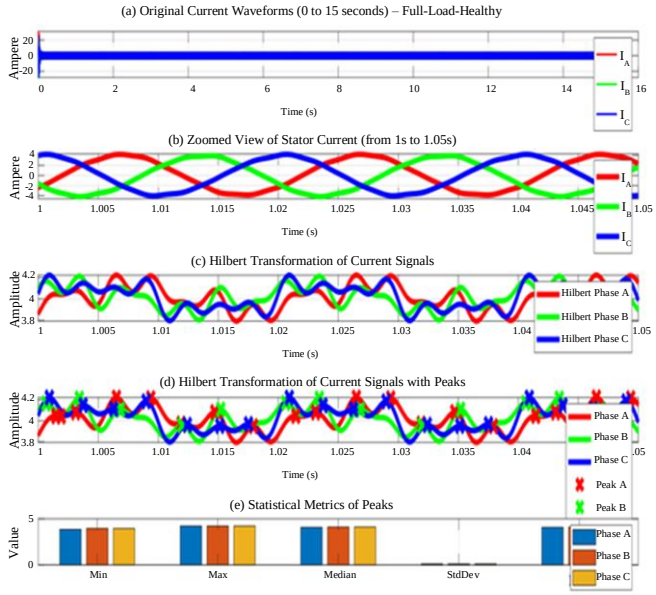
Table 1. Specifications of induction motor

S.NO	Parameters	Values
1.	Voltage	415 V
2.	Current	3 to 5A
3.	Switching Frequency	50 Hz
4.	No of Poles	4
5.	Power	1.5 kW
6.	Speed	1440

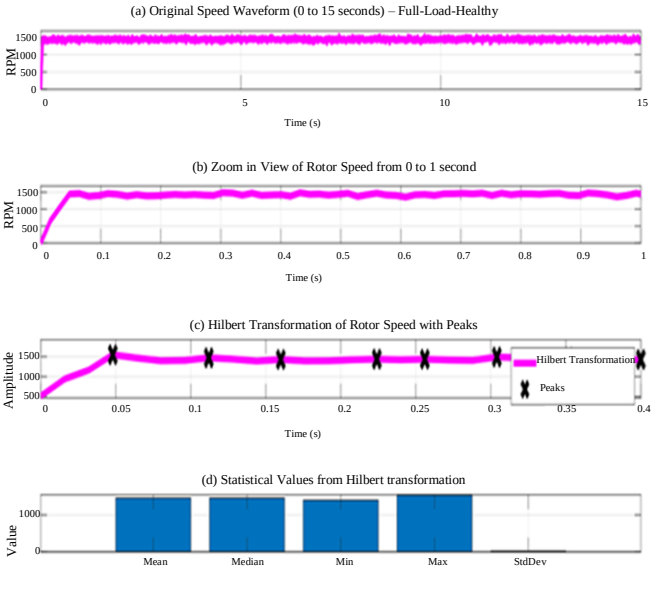
5. Results and Discussion

5.1. Hilbert Transform-Based Feature Extraction for WOPRNN

The WOAPRNN predicts motor fault conditions by extracting features from the characteristics of speed and current waveforms by using the Hilbert transform. The induction motor was analyzed under seven different fault conditions and one healthy condition (A) under all loading conditions. The seven fault conditions were: Inner Race Bearing Fault (G), Outer Race Bearing Fault (H), 1-Ball Bearing Fault (I), 2-Ball Bearing Fault (J), 1-Broken Rotor Fault (K), 2-Broken Rotor Fault (L), 10% Stator Turn Fault (M), and 25% Stator Turn Fault (N).



(a)



(b)

Fig. 13 Feature extractions of healthy induction motor at full load using Hilbert transform: (a) Current waveform, and (b) Speed waveform.

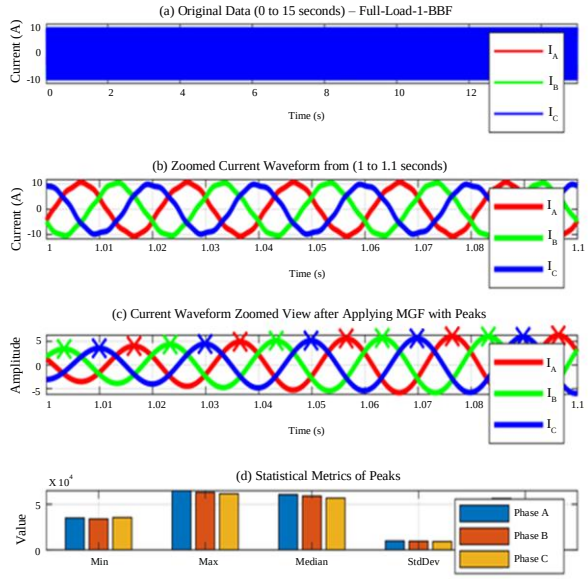
Figure 13(a) illustrates the feature extraction process from the three-phase current waveform using the Hilbert transform for a healthy induction motor under full load conditions. The characteristics of the induction motor were first captured from the experimental setup and pre-processed. A time window of 0 to 15 seconds was selected for analyzing the current characteristics of the motor, with the waveform plotted in the first row of Figure 13(a). To observe in-depth current characteristics, a zoomed-in plot of the waveform between 1 second and 1.05 seconds can be seen in the second row of Figure 13(a)The Hilbert transform was then applied to the pre-

processed current waveform, as depicted in the third section of Figure 13(a)The fourth line of Figure 13(a) displays 100 random peaks identified after applying the Hilbert transform. The next row shows a bar chart of extracted features, including minimum, maximum, standard deviation, median, and mean, for the healthy motor under full load conditions. This represents one sample. The same procedure was repeated for all seven fault conditions under various loading scenarios, and their corresponding features were extracted. Similarly, the same method was applied to extract features from the speed characteristics of the induction motor under all faulted and healthy conditions across all loading scenarios, as shown in Figure 13(b). Finally, the features extracted from the current and speed waveforms after applying the Hilbert transform were combined to form Dataset 1, which was then used to train the WOAPRNN.

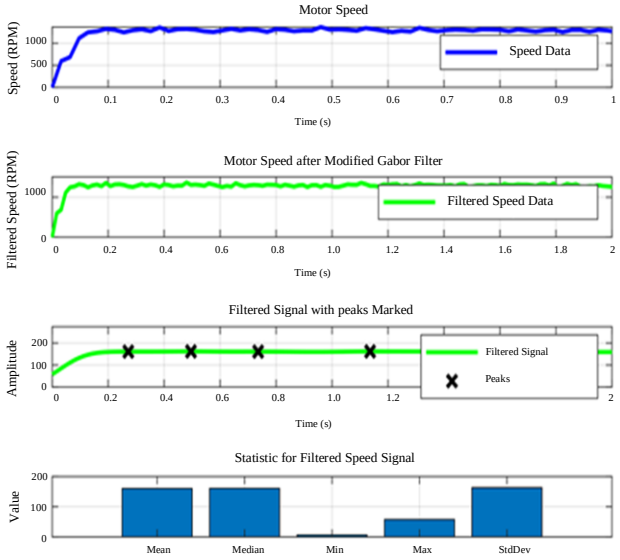
5.2. Modified Gabor Filter-Based Feature Extraction for GWOPNN

The GWOPRNN predicts motor fault conditions by extracting features from the characteristics of speed and current waveforms using the Modified Gabor Filter (MGF). The induction motor was analyzed under seven different fault conditions and one healthy condition, as mentioned earlier. Figure 14(a) illustrates the feature extraction process from the three-phase current waveform using the Modified Gabor Filter for a bearing fault with one ball broken under full load conditions. The characteristics of the induction motor were first captured from the experimental setup and pre-processed. A time window of 0 to 15 seconds was selected for analyzing the current characteristics of the motor, with the waveform plotted in the first row of Figure 14(a). To observe detailed current characteristics, a zoomed-in plot of the waveform between 1 second and 1.1 seconds is shown in the second line of Figure 14(a). The third line of Figure 14(a) displays the characteristics of the three-phase current waveform after applying the Modified Gabor Filter, highlighting the 100 random peaks identified. The fourth line of Figure 14(a) presents a bar chart of the extracted features, including minimum, maximum, standard deviation, median, and mean, for one sample of a bearing fault with one ball broken under full load conditions. The same procedure was repeated for all samples, as well as for the healthy condition and all seven fault conditions under various loading scenarios, to extract their corresponding features.

Figure 14(b) shows the speed characteristics of an induction motor with one ball broken fault under full load conditions. Similarly, the same procedure as stated earlier for feature extraction from current characteristics was carried out to extract features from the speed characteristics. The same process was repeated for all healthy and faulted conditions under various loading scenarios. Finally, the features extracted from the current and speed waveforms after applying the Modified Gabor Filter were combined to form Dataset 2, which was then used to train and test the GWOPRNN.



(a)

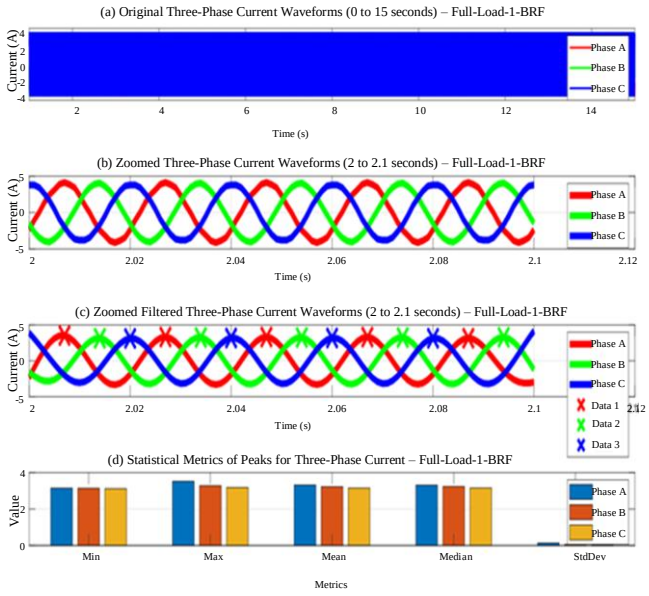


(b)

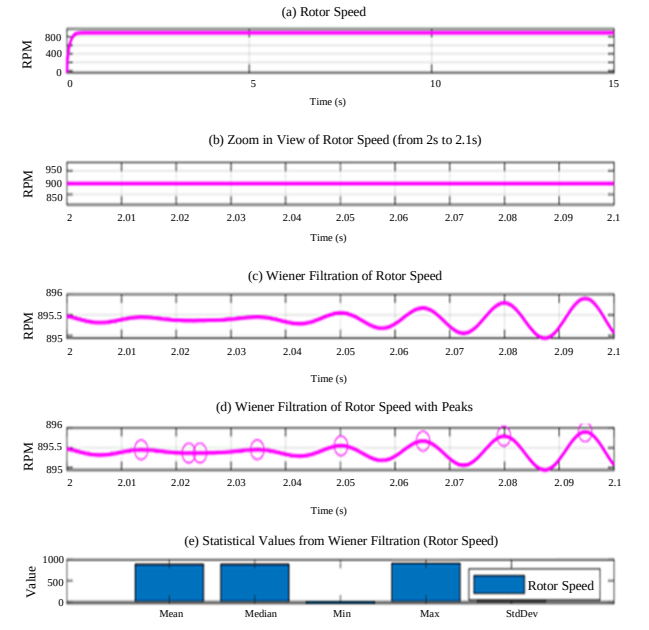
Fig. 14 Feature extraction of bearing fault with one ball broken at full load using gabor filter: (a) Current waveform, and (b) Speed waveform.

5.3. Feature Extraction by Wiener Filter for 1-D CNN

Figure 15(a) illustrates the process of extracting features from the current characteristics of an induction motor during a one-bar broken fault under full load conditions. The same procedure described in Figure 15(a) was followed to extract features from the current characteristics of the induction motor, but instead of using the Hilbert transform, the Wiener filter was applied. The corresponding features were extracted and stored for all seven faulted conditions as well as the healthy condition (A, G, H, I, J, K, L, M, N).



(a)



(b)

Fig. 15 Feature extraction of one bar broken in the rotor at full load: (a) Current waveform, and (b) Speed waveform.

Similarly, the procedure outlined in Figure 15(b) was followed to extract features from the speed characteristics of the induction motor. Instead of using the Hilbert transform, the Wiener filter was applied to the speed waveform. The extracted features were analyzed and stored for both healthy and faulted conditions across various loading scenarios.

5.4. Advantages of WOPCNN Over 1D-CNN and GWOPNN

The WOPRNN offers several notable advantages. Firstly, it provides enhanced noise filtering, as the Hilbert filter effectively reduces operational noise and vibrations, particularly under full-load conditions. Secondly, it delivers improved feature extraction capabilities, outperforming both 1D-CNN and GWOPNN in identifying intricate patterns from stator currents and rotor speeds, resulting in higher fault detection accuracy. Additionally, WOPRNN excels in detecting complex faults, consistently identifying challenging fault types such as rotor faults (K, L) and stator turn faults (M, N) with superior accuracy. Lastly, the model demonstrates consistent detection performance across various load conditions, maintaining high classification accuracy for both simple and complex fault types, which highlights its reliability and robustness.

5.5. Confusion Matrix Analysis of WOPRNN

The confusion matrices were employed to analyze the performance of the WOAPRNN by comparing its predicted outputs with the actual values. Tables 2, 3, and 4 demonstrate the classification performance of WOAPRNN for three load conditions: no load, full load, and half load, presenting the predicted outputs along with the actual outputs. Table 5 summarizes the overall classification accuracy of the WOAPRNN under all load conditions. Figures 16 to 18 depict bar charts comparing the classification performance of WOAPRNN with other neural networks, namely 1-D CNN and GWO-PNN, under all load conditions (no load, half load, and full load). Additionally, Figure 19 presents a bar chart showcasing the overall accuracy of all three neural networks. From Table 2, it was observed that WOAPRNN performed exceptionally well under fault conditions A, G, H, and I, with a slight reduction in accuracy for conditions J, K, L, and M. Similarly, Table 3 shows that WOAPRNN excelled in fault condition A but struggled in other fault conditions. Table 4 indicates strong performance in conditions A and G, with a gradual decline as other fault conditions were considered. Table 5 reveals that under no-load conditions, WOAPRNN achieved its highest accuracy of 99.28%. As the load increased, its accuracy decreased slightly to 99.06%. Despite this minor reduction, WOAPRNN achieved an impressive overall classification accuracy of 99.15%. The analysis of Figures 16 - 19 and Tables 2 - 5 demonstrates that WOAPRNN outperformed other fault classification neural networks, such as 1-D CNN and GWO-PNN, achieving the highest accuracy of 99.15%.

Table 2. WOPRNN - no load confusion matrix

Actual	A	G	H	I	J	K	L	M	N	Accuracy (%)
A	200	0	0	0	0	0	0	0	0	100.0
G	0	200	0	0	0	0	0	0	0	100.0

H	0	0	200	0	0	0	0	0	0	100.0
I	0	0	0	200	0	0	0	0	0	100.0
J	0	0	0	1	199	0	0	0	0	99.5
K	0	0	0	0	0	199	1	0	0	99.5
L	0	0	0	0	0	1	199	0	0	99.5
M	0	0	0	0	0	1	1	195	3	97.5
N	0	0	0	0	0	0	0	5	195	97.5

Table 3. WOPRNN - half load confusion matrix

Actual	A	G	H	I	J	K	L	M	N	Accuracy (%)
A	200	0	0	0	0	0	0	0	0	100
G	0	199	1	0	0	0	0	0	0	99.5
H	0	1	199	0	0	0	0	0	0	99.5
I	0	0	0	199	1	0	0	0	0	99.5
J	0	0	0	1	199	0	0	0	0	99.5
K	0	0	0	1	0	198	0	0	1	99.0
L	0	1	0	0	0	1	198	0	0	99.0
M	0	0	0	0	0	0	1	197	2	98.5
N	0	0	0	0	0	0	0	5	195	97.5

Table 4. WOPRNN - full Load Confusion Matrix

Actual	A	G	H	I	J	K	L	M	N	Accuracy (%)
A	200	0	0	0	0	0	0	0	0	100
G	0	200	0	0	0	0	0	0	0	100
H	0	1	199	0	0	0	0	0	0	99.5
I	0	0	0	199	1	0	0	0	0	99.5
J	0	0	0	1	199	0	0	0	0	99.5
K	0	0	0	1	0	197	1	0	1	98.5
L	0	1	0	1	0	0	198	0	0	99.0
M	0	0	0	0	0	1	1	196	2	98.0
N	0	0	0	0	0	0	0	5	195	97.5

Table 5. WOPRNN accuracy by fault type and load condition

Fault Type	No Load (%)	Half Load (%)	Full Load (%)	Overall Accuracy (%)
A	100.00	100.00	100.00	99.15
G	100.00	99.50	100.00	
H	100.00	99.50	99.50	
I	100.00	99.50	99.50	
J	99.50	99.50	99.50	
K	99.50	99.00	98.50	
L	99.50	99.00	99.00	
M	97.50	98.50	98.00	
N	97.50	97.50	97.50	
Average	99.28	99.11	99.06	

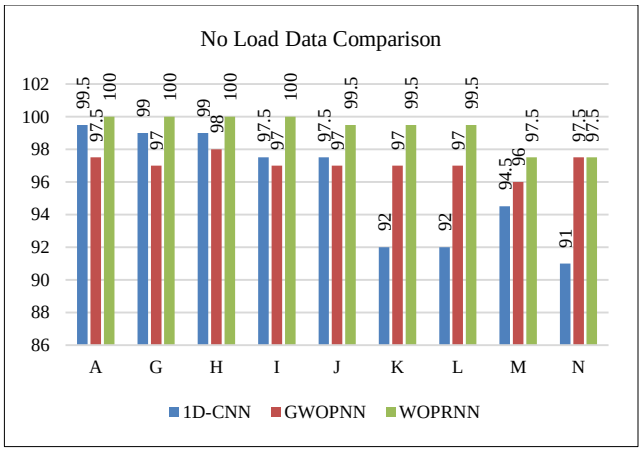


Fig. 16 No load comparison bar chart

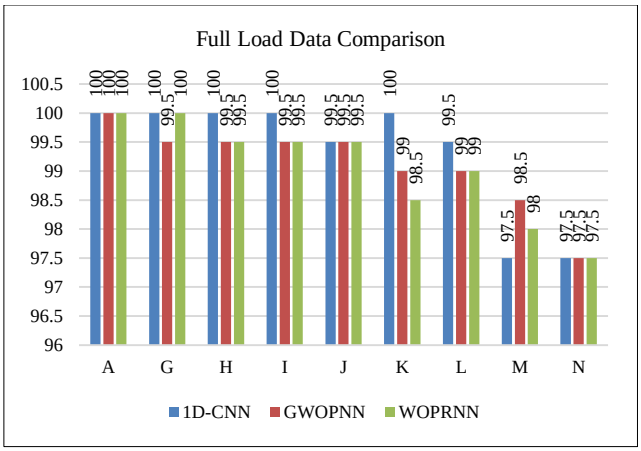


Fig. 18 Full load data bar chart

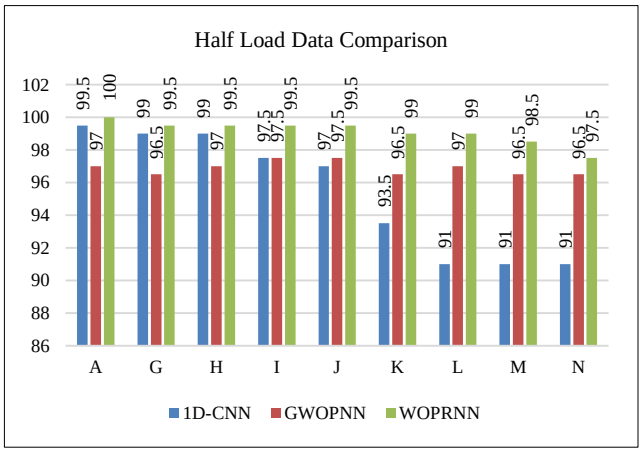


Fig. 17 Half-load comparison bar chart

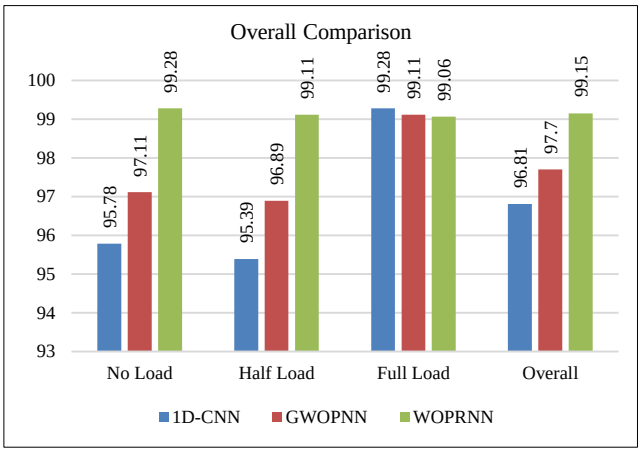


Fig. 19 Overall comparison

6. Conclusion

From experimental studies, tabulations, and bar charts, it was evident that WOAPRNN was far superior to other fault classification methods, such as 1-D CNN and GWOPRNN, for identifying faults in a 3-phase induction motor, achieving an overall accuracy of 99.15%. By considering both current and speed characteristics, this method became more accurate than most other fault classification neural networks. The work highlights the potential for enhancing predictive maintenance, reducing downtime, and improving operational efficiency, with future research opportunities in broader fault scenarios and other motor types.

References

[1] Dhiraj Neupane, and Jongwon Seok, "Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset with Deep Learning Approaches: A Review," *IEEE Access*, vol. 8, pp. 93155-93178, 2020. [CrossRef] [Google scholar][Publisher Link]

[2] Paulo Antonio Delgado-Arredondo et al., "Methodology for Fault Detection in Induction Motors Via Sound and Vibration Signals," *Mechanical Systems and Signal Processing*, vol. 83, pp. 568-589, 2017. [CrossRef] [Google scholar] [Publisher Link]

[3] Ali Namdar et al., "A Robust Stator Inter-Turn Fault Detection in Induction Motor Utilizing Kalman Filter-Based Algorithm," *Measurement*, vol. 187, 2022. [CrossRef] [Google scholar] [Publisher Link]

[4] Zuolu Wang et al., "Fault Identification of Broken Rotor Bars in Induction Motors Using an Improved Cyclic Modulation Spectral Analysis," *Energies*, vol. 12, no. 17, 2019. [CrossRef] [Google scholar] [Publisher Link]

[5] Anurag Choudhary et al., "Condition Monitoring and Fault Diagnosis of Induction Motors: A Review," *Archives of Computational Methods in Engineering*, vol. 26, pp. 1221-1238, 2019. [CrossRef] [Google scholar] [Publisher Link]

[6] Ola E. Hassan et al., "Induction Motor Broken Rotor Bar Fault Detection Techniques Based on Fault Signature Analysis-A Review," *IET Electric Power Applications*, vol. 12, no. 7, pp. 895-907, 2018. [CrossRef] [Google scholar] [Publisher Link]

[7] Majid Hussain et al., "Fault Detection and Identification Using Deep Learning Algorithms in Induction Motors," *Computer Modeling in Engineering and Sciences*, vol. 133, no. 2, pp. 435-470, 2022. [CrossRef] [Google scholar] [Publisher Link]

[8] Rafia Nishat Toma, Alexander E. Prosvirin, and Jong-Myon Kim, "Bearing Fault Diagnosis of Induction Motors Using a Genetic Algorithm and Machine Learning Classifiers," *Sensors*, vol. 20, no. 7, 2020. [CrossRef] [Google scholar] [Publisher Link]

[9] Jiangquan Zhang et al., "A New Bearing Fault Diagnosis Method Based on Modified Convolutional Neural Networks," *Chinese Journal of Aeronautics*, vol. 33, no. 2, 2019. [CrossRef] [Google scholar] [Publisher Link]

[10] Ana L. Martinez-Herrera et al., "Multiple Fault Detection in Induction Motors through Homogeneity and Kurtosis Computation," *Energies*, vol. 15, no. 4, 2022. [CrossRef] [Google scholar] [Publisher Link]

[11] Gholam Reza Agah et al., "Broken Rotor Bar and Rotor Eccentricity Fault Detection in Induction Motors Using a Combination of Discrete Wavelet Transform and Teager-Kaiser Energy Operator," *IEEE Transactions on Energy Conversion*, vol. 37, no. 3, pp. 2199-2206, 2022. [CrossRef] [Google scholar] [Publisher Link]

[12] Purushottam Gangsar, and Rajiv Tiwari, "Comparative Investigation of Vibration and Current Monitoring for Prediction of Mechanical and Electrical Faults in Induction Motor Based on Multiclass-Support Vector Machine Algorithms," *Mechanical Systems and Signal Processing*, vol. 94, pp. 464-481, 2017. [CrossRef] [Google scholar] [Publisher Link]

[13] M. Geethanjali and H. Ramadoss, "Fault Diagnosis of Induction Motors Using Motor Current Signature Analysis: A Review," *Advanced Condition Monitoring and Fault Diagnosis of Electric Machines*, pp. 1-37, 2019. [CrossRef] [Google scholar] [Publisher Link]

[14] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014. [CrossRef] [Google scholar] [Publisher Link]

[15] Yuxiang Hou et al., "Improved Grey Wolf Optimization Algorithm and Application," *Sensors*, vol. 22, no. 10, 2022. [CrossRef] [Google scholar] [Publisher Link]

[16] Jianwei Yang et al., "A Modified Gabor Filter Design Method for Fingerprint Image Enhancement," *Pattern Recognition Letters*, vol. 24, pp. 1805-1817, 2003. [CrossRef] [Google scholar] [Publisher Link]

[17] Seyedali Mirjalili, and Andrew Lewis, "The Whale Optimization Algorithm," *Advanced Engineering Software*, vol. 95, pp. 51-67, 2016. [CrossRef] [Google scholar] [Publisher Link]

[18] Zhiguang Wang, Weizhong Yan, and Tim Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," *International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, pp. 1578-1585, 2017. [CrossRef] [Google scholar] [Publisher Link]