



Robust Bearing Fault Detection and Classification Using Deep Neural Networks: A Comprehensive Study on the CWRU Dataset

Saddam Bensaoucha¹, Gharib Mousa Gharib², Maha Al Soudi³, Ali Teta⁴, Samia Benzita¹,
Ahmed Saadeddine Bertal¹, Ahlam Guiatni⁵, Abdelaziz Rabehi⁶, Mohamed Benghanem^{7*}

¹ Department of Electrical Engineering, Laboratory of Analysis and Control of Energy Systems and Electrical Networks (LACoSERE), University of Amar Telidji, Laghouat 03000, Algeria

² Department of Mathematics, Faculty of Science, Zarqa University, Zarqa 13110, Jordan

³ Department of Basic Scientific Sciences, Applied Science Private University, Amman 11931, Jordan

⁴ Department of Electrical Engineering, Applied Automation and Industrial Diagnostics Laboratory, Ziane Achour University, Djelfa 17000, Algeria

⁵ Faculty of Sciences and Technology, University of Jijel, Jijel 18000, Algeria

⁶ Laboratory of Telecommunication and Smart Systems (LTSS), Faculty of Science and Technology, University of Djelfa, Djelfa 17000, Algeria

⁷ Physics Department, Faculty of Science, Islamic University of Madinah, Madinah 42351, Saudi Arabia

Corresponding Author Email: mbenghanem@iu.edu.sa

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.581120>

ABSTRACT

Received: 1 October 2025

Revised: 24 November 2025

Accepted: 28 November 2025

Available online: 30 November 2025

Keywords:

rotating machines, bearing faults, fault detection, artificial intelligence, neural networks

This study investigates bearing fault diagnosis using the Case Western Reserve University (CWRU) dataset through a systematic approach that includes vibration signal acquisition, feature extraction, and dataset preparation. To evaluate model performance, several predefined Artificial Neural Network (ANN) architectures were trained and compared alongside an automatically optimized neural network (ONN). The predefined models provide a base-line for performance assessment, while the optimized model leverages Bayesian optimization and cross-validation to enhance generalization and reduce overfitting. Experimental results demonstrate that the ONN achieves the highest testing accuracy of 98.6% and an F1-score of 0.99, with the lowest misclassification cost, outperforming all predefined architectures, which achieved accuracies ranging from 92% to 95%. The finding of this study highlights the significance of network architecture selection and hyperparameter optimization in improving ANN-based fault detection. The proposed methodology provides a robust and efficient framework for automated bearing fault diagnosis, offering promising prospects for predictive maintenance and reliability enhancement in rotating machinery. Overall, the study confirms that combining systematic feature extraction with ONN design significantly improves diagnostic accuracy and operational reliability in industrial applications.

1. INTRODUCTION

Rotating machines, such as motors, turbines, pumps, and compressors, play a vital role in modern industry by enabling energy conversion and fluid transfer processes. Their reliable operation is critical to the efficiency and safety of industrial systems. However, these machines are subject to various electrical and mechanical faults, including misalignment, unbalance, and bearing defects, which can lead to reduced performance, unexpected shutdowns, and costly maintenance interventions. Among these, bearing faults are the most common, accounting for approximately 40-45% of rotating machine failures, followed by stator (30-37%) and rotor faults (about 10%) [1-4].

Bearings are essential mechanical elements designed to support loads while minimizing friction and wear through rolling elements such as balls or rollers. Despite their

importance, bearing defects including outer race, inner race, ball, and cage faults remain a major source of vibration, noise, overheating, and mechanical instability. If undetected, these issues can escalate to severe damage, system downtime, and significant economic losses. Consequently, the timely detection and diagnosis of bearing faults have become a cornerstone of predictive maintenance strategies [4-9].

Traditional approaches to fault detection, such as vibration analysis, thermography, and ultrasonic testing, provide valuable diagnostic information but often rely on human interpretation and predefined fault signatures [10-14]. These limitations have motivated the growing adoption of data-driven methods, particularly machine learning algorithms, for automated fault detection and classification. Neural networks, in particular, have shown strong potential in modeling nonlinear patterns in vibration signals and improving diagnostic accuracy [15-23].

This study investigates the application of Multilayer Perceptron (MLP) neural networks for the detection and classification of bearing faults in rotating machinery. Vibration signals are analyzed, and statistical features such as maximum, minimum, and median values are extracted as network inputs. Several architectures are evaluated using benchmark datasets, including the Case Western Reserve University (CWRU) dataset.

- The contributions of this paper are threefold:
- To highlight the significance of bearing fault detection in industrial rotating machinery.
 - To evaluate the effectiveness of MLP neural networks for fault classification using vibration features.
 - To propose an automated framework that improves accuracy and reliability in predictive maintenance.

The remainder of this paper is organized as follows: Section 2 briefly presents the architecture of neural networks; Section 3 describes the CWRU dataset; Section 4 introduces the application of neural networks for bearing fault detection; Section 5 discusses the classification results; and Section 6 concludes the study.

2. NEURAL NETWORKS FOR FAULT DETECTION

Artificial Neural Networks (ANNs), illustrated in Figure 1, are computational models inspired by the structure and functioning of biological neurons in the human brain. They are composed of interconnected layers of processing units (neurons) organized into an input layer, one or more hidden layers, and an output layer [10, 22-26].

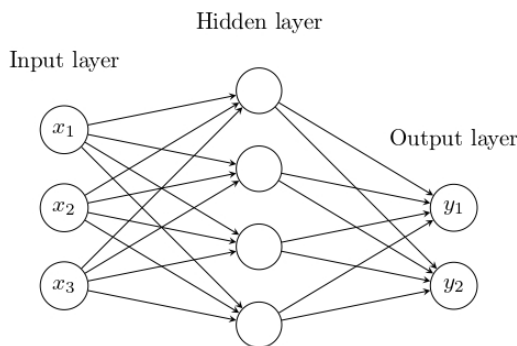


Figure 1. Schematic diagram of the neural network

In the context of classification tasks, the input layer receives raw data features, which are then processed through hidden layers using weighted connections and nonlinear activation functions. These transformations enable the network to capture complex relationships and extract relevant patterns from the data. The output layer generates the final decision, typically represented as a probability distribution over the possible classes, from which the most likely class is selected.

Due to their capacity to model nonlinear relationships and handle high-dimensional data, ANNs are particularly effective for pattern recognition, classification, and prediction problems. In this study, they are of significant importance for analyzing complex datasets and enhancing the reliability of decision-making processes.

In the context of bearing fault diagnosis, ANNs can learn the complex nonlinear relationships between the measured signals (such as vibration or current) and the fault types. Once trained, the ANN can classify new unseen data with high

accuracy [27-30].

3. CWRU DATASET DESCRIPTION

The dataset utilized in this study is sourced from the CWRU bearing data center, which is widely regarded as one of the most comprehensive and frequently cited benchmarks in the field of bearing fault diagnosis [27, 28]. The experiments were carried out using a 2-horsepower reliance electric motor, in which faults were artificially induced using electro-discharge machining as shown in Figure 2.

The dataset provides vibration signal data collected under both healthy and faulty bearing conditions, encompassing four distinct states: healthy or faulty bearings as illustrated in Figure 3.

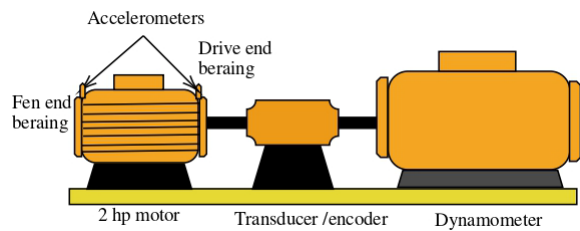


Figure 2. Experimental setup of the CWRU bearing fault test stand

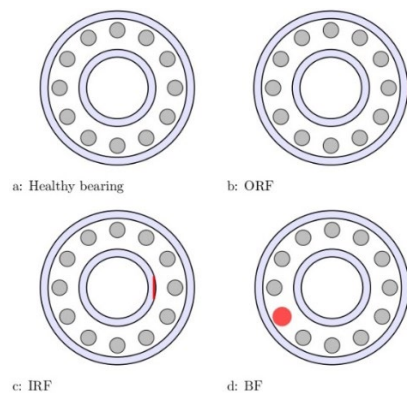


Figure 3. Cross-sectional diagrams of a radial ball bearing under different conditions

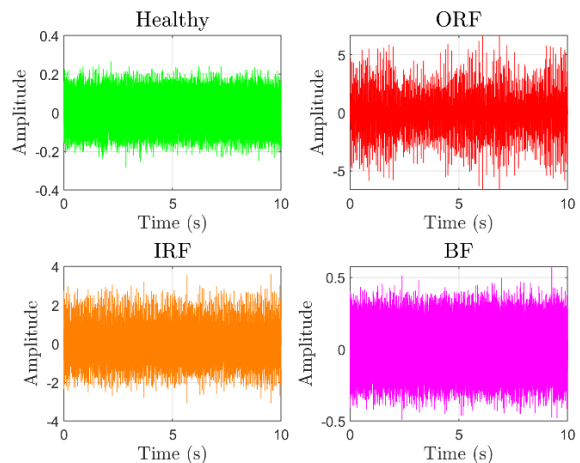


Figure 4. Vibration signals for healthy and faulty bearings under 3 hp load condition. Healthy (green), ORF (red), IRF (orange), BF (magenta)

The dataset includes fault sizes ranging from 0.007 to 0.028 inches. Vibration signals were captured using accelerometers mounted both near and far from the drive-end bearing. Measurements were taken under varying motor loads (from 0 to 3 horsepower) and speeds ranging from 1720 to 1797 RPM. Figure 4 illustrates the vibration signals acquired under a 3 hp load condition for different bearing states. It can be observed that the signal amplitude differs noticeably among the cases.

Table 1. CWRU bearing dataset parameters

Fault Type	Fault Size (in)	Load (hp)	Speed (RPM)	Data Format
Healthy	0	0, 1, 2, 3	1720–1797	.mat
ORF	0.007, 0.014, 0.021	0, 1, 2, 3	1720–1797	.mat
IRF	0.007, 0.014, 0.021, 0.028	0, 1, 2, 3	1720–1797	.mat
BF	0.007, 0.014, 0.021, 0.028	0, 1, 2, 3	1720–1797	.mat

4. NEURAL NETWORKS FOR BEARING FAULT DETECTION

To assess model performance, five predefined feedforward ANN architectures of varying depth and neuron count were trained alongside one automatically ONN. This setup allowed for a comparison of standard network configurations with an optimized architecture leveraging hyperparameter tuning for improved accuracy and generalization.

Figure 5 outlines the bearing fault diagnosis pipeline, from raw vibration signal acquisition to final classification.

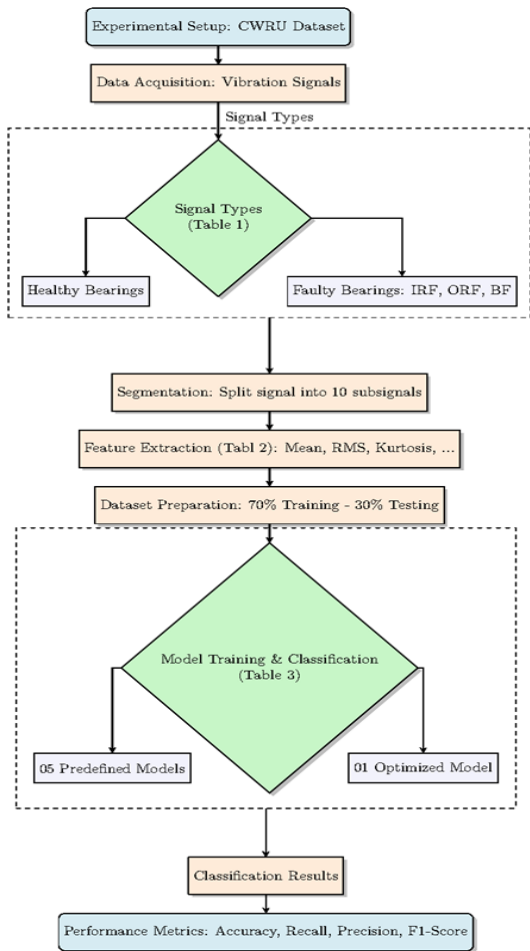


Figure 5. Workflow of the bearing fault diagnosis methodology using neural networks

These acceleration signals serve as the primary input for training and validating the machine learning models developed in this study, particularly the MLP, to facilitate accurate detection and classification of bearing fault types [31-34].

Table 1 presents the parameters of the CWRU bearing dataset, including fault types, fault sizes, load conditions, rotational speeds, and the corresponding data format.

4.1 Data pre-processing and feature extraction

In this study, vibration signals were extracted from the CWRU dataset under four operating conditions: healthy bearing, BF, IRF, and ORF. For each condition, the full-length signal was segmented into 10 equal-length sub signals to ensure a balanced and diverse dataset.

Figure 6 shows the segmented vibration signals for the healthy bearing and the three faulty conditions (ORF, IRF, BF) under a 3 hp load. The signals are divided into ten subsignals.

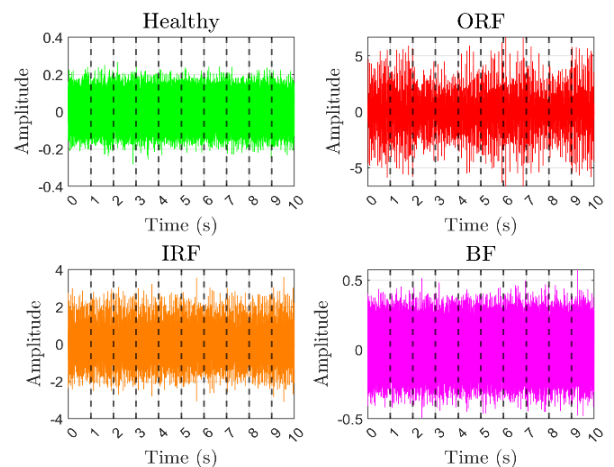


Figure 6. Segmented vibration signals for healthy and faulty bearings under a 3 hp load condition. Healthy (green), ORF (red), IRF (orange), and BF (magenta)

Each signal has a duration of 10 s and was sampled at a frequency of 12 kHz, resulting in 120000 data points per signal. The signals are divided into 10 sub signals.

This segmentation process not only highlights the amplitude differences among bearing conditions but also serves to augment the dataset, thereby providing a larger and more representative input set for training the neural network.

Table 2 summarizes the structure of the final processed dataset used for training the neural network models. For each bearing class, the table shows the number of original signals, their composition in terms of sizes and loads, the total number of sub signals obtained after segmentation, the number of features extracted per sub signal, and the resulting total feature samples. As shown, segmentation and feature extraction expand the dataset to 4800 samples, ensuring sufficient representation of all bearing conditions for accurate ANN

training and evaluation.

The extracted statistical features and their corresponding mathematical expressions are summarized in Table 3. All ten of these features were utilized for training the classifiers

without applying any further feature selection. Prior to training, each feature was normalized to a [0, 1] range using min-max scaling to ensure equal contribution during the learning process (Appendix A).

Table 2. Summary of common ANN models and their use cases

Class	No of. Original Signals	Comp.	No. of Sub-signs	Features Per Sub-signal	Total Feature Samples
Healthy	4	4 loads	$10 \times 4 = 40$	10	400
ORF	12	3 sizes \times 4 loads	$10 \times 12 = 120$	10	1200
IRF	16	4 sizes \times 4 loads	$10 \times 16 = 160$	10	1600
BF	16	4 sizes \times 4 loads	$10 \times 16 = 160$	10	1600
Total	48	—	480	10	4800

Table 3. Extracted statistical features and their mathematical expressions

Feature Name	Symbol	Mathematical Expression
Minimum	$\min(x)$	$\text{Min}(x_1, x_2, \dots, x_N)$
Maximum	$\max(x)$	$\text{Max}(x_1, x_2, \dots, x_N)$
Mean	μ	$\mu = (1/N) \sum x_i$
Median	Med	Middle value of ordered set
Root Mean Square	RMS	$\text{RMS} = \sqrt{(1/N) \sum x_i^2}$
Standard Deviation	Σ	$\sigma = \sqrt{(1/N) \sum (x_i - \mu)^2}$
Kurtosis	Kur	$\text{kur} = (1/N) \sum ((x_i - \mu)/\sigma)^4$
Peak Value	Peak	$\text{Max}(x_i)$
Crest Factor	CF	$\text{CF} = \text{Peak} / \text{RMS}$
Impulse Factor	IF	$\text{IF} = \text{Peak} / [(1/N) \sum x_i]$

4.2 Training and validation

The dataset was randomly divided into two subsets: 70% for training and 30% for testing. To enhance generalization and prevent overfitting, a 5-fold cross-validation strategy was applied during training [35-40].

4.3 Neural network models

Six ANN architectures were evaluated for bearing fault classification, including five predefined feedforward networks of varying depth and neuron count, along with one optimized network model. The optimized network was automatically tuned to achieve an optimal balance between classification accuracy and computational complexity.

Hyperparameters are predefined configuration parameters that govern the learning process of a neural network and are set prior to training. They determine the structure and behavior of the model, including the number of hidden layers, the number of neurons per layer, and the type of activation function (ReLU, sigmoid, or tanh). All models, including the optimized network, employed cross-validation to improve generalization and ensure robust performance (Appendix B). The optimized network further leverages automatic hyperparameter tuning, utilizing techniques such as Bayesian optimization to enhance predictive reliability (Appendix C).

The evaluated models include the Narrow Neural Network (NNN), Medium Neural Network (MNN), Wide Neural Network (WNN), Bilayered Neural Network (BNN), Trilayered Neural Network (TNN), and the Optimizable Neural Network (ONN). All networks were trained using the backpropagation algorithm.

Table 4 presents a summary of common ANN models, detailing their structural characteristics and typical use cases.

Table 4. Summary of common ANN models and their use cases

Model	Definition	Use Case
NNN	Few neurons per hidden layer	Simple problems or limited data
MNN	Moderate number of neurons per layer, between narrow and wide	General applications balancing performance and cost
WNN	Large number of neurons per hidden layer	Complex problems with large datasets
BNN	Two hidden layers	Moderate complexity tasks requiring additional depth
TNN	Three hidden layers	Complex problems such as signal processing or image recognition
ONN	Architecture or hyperparameters optimized automatically (Bayesian optimization)	Tasks requiring high accuracy and adaptive tuning

4.4 Performance evaluation metrics

To comprehensively assess the performance of classification models, several standard metrics are considered, including accuracy, recall (Rec), Precision (Pre), and F1-score. These metrics, computed from the confusion matrix components—True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)—provide complementary insights into model performance. Using them together ensures a balanced evaluation, capturing both the model's overall correctness and its ability to correctly identify

positive and negative instances, which is particularly important in cases of imbalanced datasets. Eqs. (1)-(4) present the formulas for accuracy, precision, recall, and the F1-score, respectively [26, 41-43].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F1-Score = 2 \times \frac{Precision + Recall}{Precision + Recall} \quad (4)$$

In addition to the aforementioned metrics, the total misclassification cost is reported in the results. This cost was calculated using a uniform cost matrix where any prediction error is assigned a unit penalty. Specifically, for our four-class problem, a cost of 0 was applied for correct classifications and a cost of 1 was applied for any misclassification. Consequently,

the reported 'Cost' is equivalent to the total number of misclassified samples in the dataset.

5. CLASSIFICATION RESULTS

The performance of the six ANN architectures (NNN, MNN, WNN, BNN, TNN, and ONN) was assessed during training and testing. Table 5 summarizes classification accuracy and total misclassification cost, while Table 6 reports recall, precision, and F1-score. The hyperparameter settings of each model are provided in Table 7.

Table 5. Classification accuracy and total cost of ANN models

Model	Training Phase		Testing Phase	
	Accuracy	Cost	Accuracy	Cost
NNN	94.3%	19	96.5%	5
MNN	95.8%	14	94.4%	8
WNN	94.3%	19	95.8%	6
BNN	95.8%	14	96.5%	5
TNN	93.8%	21	97.9%	3
ONN	97.6%	8	98.6%	2

Table 6. Performance evaluation of ANN models based on recall, precision, and F1-score

Model	Training Phase			Testing Phase		
	Rec	Pre	F1-Score	Rec	Pre	F1-Score
NNN	0.95	0.95	0.95	0.99	0.98	0.97
MNN	0.95	0.96	0.96	0.94	0.96	0.95
WNN	0.94	0.93	0.94	0.96	0.97	0.96
BNN	0.95	0.96	0.96	0.97	0.96	0.96
TNN	0.94	0.92	0.93	0.98	0.99	0.98
ONN	0.97	0.98	0.97	0.99	0.99	0.99

Table 7. Hyperparameter settings of ANN models

Model	No. of Layers	First Layer	Second Layer	Third Layer	Activation Function
NNN	1	10	–	–	ReLU
MNN	1	25	–	–	ReLU
WNN	1	100	–	–	ReLU
BNN	2	10	10	–	ReLU
TNN	3	10	10	10	ReLU
ONN	1	12	–	–	Sigmoid

These results form the basis for the following discussion.

5.1 Discussion of results

The results indicate that the ONN consistently outperformed all other models, achieving the highest testing accuracy (98.6%), an F1-score of 0.99, and the lowest misclassification cost (2). This confirms the advantage of automatic hyperparameter tuning, such as Bayesian optimization and cross-validation, in enhancing generalization and predictive reliability.

The TNN provided the next best performance (97.9% accuracy, F1-score 0.98), but with greater computational cost, highlighting those deeper architectures improve accuracy at the expense of complex ity. The MNN and WNN achieved moderate results, reflecting a balance between simplicity and performance, while NNN and BNN remained less effective compared with deeper or optimized architectures.

Most models obtained recall, precision, and F1-scores around 0.95 in training, suggesting stable learning. However,

ONNs consistently superior testing performance demonstrates stronger robustness and reduced overfitting.

In summary, the ONN emerges as the most reliable architecture for bearing fault diagnosis, combining high accuracy, low cost, and optimized parameters. These findings emphasize the importance of architectural and hyperparameter optimization in ANN-based fault detection.

It is worth emphasizing that, unlike many previous studies that selectively use data subsets to artificially boost classification accuracy, this study employed the entire CWRU dataset without any selective filtering. This comprehensive approach provides a more realistic evaluation of the models' performance across all fault types and operating conditions, thereby enhancing the validity and generalizability of our findings.

At the same time, it should be acknowledged that this study was conducted solely using the CWRU dataset. While the results are encouraging, the generalizability of the models to other bearing types or to real-world industrial environments with different noise levels and operational conditions remains

to be validated. Future work will focus on evaluating these architectures on additional datasets to assess their robustness and practical applicability.

6. CONCLUSION

This study presented a systematic approach for bearing fault diagnosis using vibration signals from the CWRU dataset. Several predefined ANN architectures were evaluated alongside an automatically optimized neural network (ONN) to assess the impact of network design and hyperparameter tuning on classification performance.

The experimental results demonstrated that the ONN consistently outperformed all predefined models, achieving the highest testing accuracy of 98.6%, an F1-score of 0.99, and the lowest misclassification cost. These findings confirm the synergistic effect of combining systematic feature extraction with automatic hyperparameter optimization. The statistical features provide a compact, discriminative representation of the raw vibration signals, which simplifies the learning task. Bayesian optimization then tailors the network architecture specifically to this feature set, finding the optimal depth, width, and training parameters to maximize information extraction and minimize overfitting. This synergy is evident in the ONN's superior generalization and robustness compared to models where architecture and features were not co-optimized.

Deeper architectures, such as the TNN, also achieved high accuracy but incurred greater computational costs, highlighting the trade-off between complexity and performance. Predefined models with moderate depth or width yielded reasonable results, yet they were surpassed by the ONN in both accuracy and robustness.

Overall, this work emphasizes the critical role of integrating network architecture selection and hyperparameter tuning with feature engineering for ANN-based fault diagnosis. The proposed methodology provides a robust and efficient framework for automated bearing fault detection, supporting predictive maintenance strategies and enhancing operational reliability in rotating machinery. These results confirm that ONN design, when driven by informative features, offers significant and reliable improvements for industrial diagnostic applications.

ACKNOWLEDGMENT

The researchers wish to extend their sincere gratitude to the Deanship of Scientific Research at the Islamic University of Madinah (KSA) for the support provided to the Post-Publishing Program.

The authors also acknowledge Zarqa University (Jordan) for their support and contributions to this research.

REFERENCES

- [1] Thomson, W.T., Fenger, M. (2001). Current signature analysis to detect induction motor faults. *IEEE Industry Applications Magazine*, 7(4): 26-34. <https://doi.org/10.1109/2943.930988>
- [2] Benbouzid, M.E.H., Kliman, G.B. (2003). What stator current processing-based technique to use for induction motor rotor faults diagnosis? *IEEE Transactions on Energy Conversion*, 18(2): 238-244. <https://doi.org/10.1109/TEC.2003.811741>
- [3] Zhang, P., Du, Y., Habetler, T.G., Lu, B. (2010). A survey of condition monitoring and protection methods for medium-voltage induction motors. *IEEE Transactions on Industry Applications*, 47(1): 34-46. <https://doi.org/10.1109/TIA.2010.2090839>
- [4] Riera-Guasp, M., Antonino-Daviu, J.A., Capolino, G.A. (2014). Advances in electrical machine, power electronic, and drive condition monitoring and fault detection: State of the art. *IEEE Transactions on Industrial Electronics*, 62(3): 1746-1759. <https://doi.org/10.1109/TIE.2014.2375853>
- [5] Li, X., Yang, Y., Pan, H., Cheng, J., Cheng, J. (2019). A novel deep stacking least squares support vector machine for rolling bearing fault diagnosis. *Computers in Industry*, 110: 36-47. <https://doi.org/10.1016/j.compind.2019.05.005>
- [6] Smith, W.A., Randall, R.B. (2015). Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mechanical Systems and Signal Processing*, 64: 100-131. <https://doi.org/10.1016/j.ymssp.2015.04.021>
- [7] Peng, B., Bi, Y., Xue, B., Zhang, M., Wan, S. (2022). A survey on fault diagnosis of rolling bearings. *Algorithms*, 15(10): 347. <https://doi.org/10.3390/a15100347>
- [8] Wu, G., Yan, T., Yang, G., Chai, H., Cao, C. (2022). A review on rolling bearing fault signal detection methods based on different sensors. *Sensors*, 22(21): 8330. <https://doi.org/10.3390/s22218330>
- [9] Gangsar, P., Tiwari, R. (2020). Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. *Mechanical Systems and Signal Processing*, 144: 106908. <https://doi.org/10.1016/j.ymssp.2020.106908>
- [10] Bensaoucha, S., Bessedik, S.A., Ameer, A., Teta, A. (2019). Induction motors broken rotor bars detection using RPVM and neural network. *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 38(2): 596-615. <https://doi.org/10.1108/COMPEL-06-2018-0256>
- [11] Elbouchikhi, E., Choqueuse, V., Amirat, Y., Benbouzid, M.E.H., Turri, S. (2017). An efficient Hilbert–Huang transform-based bearing faults detection in induction machines. *IEEE Transactions on Energy Conversion*, 32(2): 401-413. <https://doi.org/10.1109/TEC.2017.2661541>
- [12] Puche-Panadero, R., Pineda-Sanchez, M., Riera-Guasp, M., Roger-Folch, J., Hurtado-Perez, E., Perez-Cruz, J. (2009). Improved resolution of the MCSA method via Hilbert transform, enabling the diagnosis of rotor asymmetries at very low slip. *IEEE Transactions on Energy Conversion*, 24(1): 52-59. <https://doi.org/10.1109/TEC.2008.2003207>
- [13] Georgoulas, G., Loutas, T., Stylios, C.D., Kostopoulos, V. (2013). Bearing fault detection based on hybrid ensemble detector and empirical mode decomposition. *Mechanical Systems and Signal Processing*, 41(1-2): 510-525. <https://doi.org/10.1016/j.ymssp.2013.02.020>
- [14] Ghabar, I., Burqan, A., Gharib, G. (2024). The optical model absorption term in the frame of fractional derivatives. *Atoms*, 12(7): 37. <https://doi.org/10.3390/atoms12070037>
- [15] Chennana, A., Megherbi, A.C., Bessous, N., Sbaa, S., et

- al. (2025). Vibration signal analysis for rolling bearings faults diagnosis based on deep-shallow features fusion. *Scientific Reports*, 15(1): 9270. <https://doi.org/10.1038/s41598-025-93133-y>
- [16] Chennana, A., Bessous, N., Megherbi, A.C., Saidi, L., Sbaa, S., Sayadi, M., Teta, A. (2025). Deep-shallow features fusion for induction motor fault diagnosis through infrared thermography images. In 2025 International Conference on Control, Automation and Diagnosis (ICCAD), Barcelona, Spain, pp. 1-6. <https://doi.org/10.1109/ICCAD64771.2025.11099458>
- [17] Bensaoucha, S., Bessedik, S.A., Ameer, A., Moreau, S., Teta, A. (2020). A comparative study for broken rotor bars fault detection in induction machine using DWT and MUSIC techniques. In 2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP), El Oued, Algeria, pp. 523-528. <https://doi.org/10.1109/CCSSP49278.2020.9151772>
- [18] Shendi, A.F., Nahhas, A.M. (2023). Power generation by using Photovoltaic systems for Yanbu and Rabigh regions in Saudi Arabia: A cost-effective study. *The Islamic University Journal of Applied Sciences (IUJAS)*, 2023(7): 10-44. <https://doi.org/10.63070/jesc.2023.001>
- [19] Alrehaily, A. (2024). Advances in bioinformatics techniques to predict neoantigen: Exploring tumor immune microenvironment and transforming data into therapeutic insights. *The Islamic University Journal of Applied Sciences (IUJAS)*, 2024(12): 81-108. <https://doi.org/10.63070/jesc.2024.016>
- [20] Yang, T., Huang, S. (2017). Fault diagnosis based on improved deep belief network. In 2017 5th International Conference on Enterprise Systems (ES), Beijing, China, pp. 305-310. <https://doi.org/10.1109/ES.2017.57>
- [21] Xia, M., Li, T., Xu, L., Liu, L., De Silva, C.W. (2017). Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks. *IEEE/ASME transactions on mechatronics*, 23(1): 101-110. <https://doi.org/10.1109/TMECH.2017.2728371>
- [22] Konar, P., Chattopadhyay, P. (2011). Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs). *Applied Soft Computing*, 11(6): 4203-4211. <https://doi.org/10.1016/j.asoc.2011.03.014>
- [23] Alsaawy, Y. (2024). Machine learning pipeline: Feature selection and adaptive training for DDoS detection to improve cloud security. *The Islamic University Journal of Applied Sciences (IUJAS)*, 5(2): 242-272. <https://doi.org/10.63070/jesc.2024.023>
- [24] Siddique, M.F., Zaman, W., Ullah, S., Umar, M., et al. (2024). Advanced bearing-fault diagnosis and classification using mel-scalograms and FOX-optimized ANN. *Sensors*, 24(22): 7303. <https://doi.org/10.3390/s24227303>
- [25] Chen, X., Li, J., Yu, A., Cai, B., Wu, Q., Xia, M. (2025). Ultra-low latency ANN-SNN conversion for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 74: 1-10. <https://doi.org/10.1109/TIM.2025.3548254>
- [26] Zhang, S., Zhang, S., Wang, B., Habetler, T.G. (2020). Deep learning algorithms for bearing fault diagnostics—A comprehensive review. *IEEE Access*, 8: 29857-29881. <https://doi.org/10.1109/ACCESS.2020.2972859>
- [27] Bakria, D., Beladel, A., Korich, B., Teta, A., et al. (2025). A novel enhanced Grey Wolf Optimizer for global optimization problems: Application to photovoltaic parameter extraction. *Energy Reports*, 14: 2782-2796. <https://doi.org/10.1016/j.egyr.2025.09.027>
- [28] Tibermacine, I.E., Russo, S., Scarano, G., Tedesco, G., et al. (2025). Conditional VAE for personalized neurofeedback in cognitive training. *PLoS One*, 20(10): e0335364. <https://doi.org/10.1371/journal.pone.0335364>
- [29] Teta, A., Elbar, M., Beladel, A., Bakria, D., Korich, B., Chennana, A., Rabehi, A. (2025). Robust and fast fault diagnosis framework for grid-connected photovoltaic systems via adaptive channel-wise representations and transfer learning. *Measurement*, 258: 119290. <https://doi.org/10.1016/j.measurement.2025.119290>
- [30] Mohamed, A., Nacera, Y., Ahcene, B., Teta, A., et al. (2025). Optimized YOLO based model for photovoltaic defect detection in electroluminescence images. *Scientific Reports*, 15(1): 32955. <https://doi.org/10.1038/s41598-025-13956-7>
- [31] Ferkous, K., Menakh, S., Guermoui, M., Bellaour, A., et al. (2025). Optimized solar power forecasting: A multi-decomposition framework using VMD and swarm techniques. *AIP Advances*, 15(9). <https://doi.org/10.1063/5.0282210>
- [32] Taibi, A., Ikhlaf, N., Aomar, L., Touati, S., et al. (2025). Diagnosis of misalignment faults using the DTCWT-RCMFDE and LSSVM algorithms. *Scientific Reports*, 15(1): 32128. <https://doi.org/10.1038/s41598-025-12407-7>
- [33] Ali, M., Souahlia, A., Rabehi, A., Guermoui, M., et al. (2025). A robust deep learning approach for photovoltaic power forecasting based on feature selection and variational mode decomposition. *Journal of the Nigerian Society of Physical Sciences*, 7(3): 2795. <https://doi.org/10.46481/jnsps.2025.2795>
- [34] Ali, M., Rabehi, A., Souahlia, A., Guermoui, M., et al. (2025). Enhancing PV power forecasting through feature selection and artificial neural networks: A case study. *Scientific Reports*, 15(1): 22574. <https://doi.org/10.1038/s41598-025-07038-x>
- [35] Ouahabi, M.S., Benyounes, A., Barkat, S., Ihammouchen, S., et al. (2025). Real-time sensor fault tolerant control of DC-DC converters in DC microgrids using a switching unknown input observer. *IEEE Access*, 13: 95837-95850. <https://doi.org/10.1109/ACCESS.2025.3571650>
- [36] Ladjal, B., Nadour, M., Bechouat, M., Hadroug, N., et al. (2025). Hybrid deep learning CNN-LSTM model for forecasting direct normal irradiance: A study on solar potential in Ghardaia, Algeria. *Scientific Reports*, 15(1): 15404. <https://doi.org/10.1038/s41598-025-94239-z>
- [37] Bentegri, H., Rabehi, M., Kherfane, S., Nahool, T.A., et al. (2025). Assessment of compressive strength of eco-concrete reinforced using machine learning tools. *Scientific Reports*, 15(1): 5017. <https://doi.org/10.1038/s41598-025-89530-y>
- [38] Mehallou, A., M'hamdi, B., Amari, A., Tegar, M., et al. (2025). Optimal multiobjective design of an autonomous hybrid renewable energy system in the Adrar Region, Algeria. *Scientific Reports*, 15(1): 4173. <https://doi.org/10.1038/s41598-025-88438-x>
- [39] Ladjal, B., Tibermacine, I.E., Bechouat, M., Sedraoui, M., et al. (2024). Hybrid models for direct normal irradiance forecasting: A case study of Ghardaia zone (Algeria). *Natural Hazards*, 120(15): 14703-14725.

- <https://doi.org/10.1007/s11069-024-06837-1>
- [40] Baitiche, O., Bendelala, F., Cheknane, A., Rabehi, A., Comini, E. (2024). Numerical modeling of hybrid solar/thermal conversion efficiency enhanced by metamaterial light scattering for ultrathin PbS QDs-STPV cell. *Crystals*, 14(7): 668. <https://doi.org/10.3390/cryst14070668>
- [41] Hamadneh, T., Batiha, B., Gharib, G.M., Aribowo, W. (2025). Application of orangutan optimization algorithm for feature selection problems. *INASS Express*, 1(1): 1-9. <https://doi.org/10.22266/inassexpress.2025.001>
- [42] Neupane, D., Seok, J. (2020). Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access*, 8: 93155-93178. <https://doi.org/10.1109/ACCESS.2020.2990528>
- [43] Alsaoudi, M., Gharib, G.M., Al-Husban, A., Abudayyeh, J.A. (2026). A unified framework for solving abel's and linear volterra integral equations and their neutrosophic generalizations using the GALM transform. *International Journal of Neutrosophic Science (IJNS)*, 27(1): 19-35. <https://doi.org/10.54216/IJNS.270103>

NOMENCLATURE

CWRU	Case Western Reserve University
MLP	Multilayer Perceptron
ANN	Artificial Neural Network
IRF	Inner Race Fault
ORF	Outer Race Fault
BF	Ball Fault
NNN	Narrow Neural Network
MNN	Medium Neural Network
WNN	Wide Neural Network
BNN	Bilayered Neural Network
TNN	Trilayered Neural Network
ONN	Optimizable Neural Network
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

APPENDIX

Appendix A. Feature extraction code and data splitting

The following MATLAB code was used to extract the ten statistical features from each signal segment. The same logic was repeated for every bearing condition in the dataset.

```
% --- Load Data and Initialize Parameters ---
load('97.mat'); % Load the specific data file
signal = X097_DE_time(1:120000,:);
N = length(signal);
num_segments = 10;
segment_length = floor(N / num_segments);

% Preallocate results matrix
results = zeros(num_segments, 10);

% --- Calculate Features for Each Segment ---
for i = 1:num_segments
    % Define segment indices
```

```
start_idx = (i-1) * segment_length + 1;
if i == num_segments
    end_idx = N; % Ensure last segment captures all
remaining points
else
    end_idx = i * segment_length;
end

% Extract the signal segment
sub_signal = signal(start_idx:end_idx);

% Calculate the 10 statistical features
min_val = min(sub_signal);
max_val = max(sub_signal);
mean_val = mean(sub_signal);
median_val = median(sub_signal); % Using built-in
function for robustness
rms_val = sqrt(mean(sub_signal.^2));
std_val = std(sub_signal);
kurtosis_val = kurtosis(sub_signal);
peak_val = max(abs(sub_signal));
crest_factor = peak_val / rms_val;
impulse_factor = peak_val / mean(abs(sub_signal));

% Store results for the segment
results(i, :) = [min_val, max_val, mean_val,
median_val, rms_val, ...
std_val, kurtosis_val, peak_val,
crest_factor, impulse_factor];
end

% --- Save Features to File ---
feature_names = {'Min', 'Max', 'Mean', 'Median', 'RMS',
'StdDev', ...
'Kurtosis', 'Peak', 'CrestFactor',
'ImpulseFactor'};
T = array2table(results, 'VariableNames',
feature_names);
writetable(T, 'features_DE.xlsx');
disp('Features successfully saved to
"features_DE.xlsx");
```

Appendix B.

This appendix contains the core MATLAB code template used to train and evaluate all ANN models presented in the paper. The specific hyperparameters for each model (NNN, MNN, WNN, BNN, TNN, ONN)—including the number of layers, neurons per layer, and activation function—were varied according to the configurations specified in Table 7.

```
function [resultsTable, metricsTable] =
NN_model(train_data, test_data)
% Neural Network classifier with performance metrics
% Extract predictors and response
predictorNames = {'Max', 'Min', 'Median', 'Std', 'RMS',
'Kurtosis', 'Skewness', 'Range', 'CF', 'IF'};
predictors = train_data(:, predictorNames);
response = train_data.Target;
% Train neural network
classificationNeuralNetwork = fitnet(predictors,
response, ...
'LayerSizes', [85 80], 'Activations', 'relu', ...
'Lambda', 2.230905500034876e-08, 'IterationLimit',
1000, ...
```

```

'Standardize', True);
% Cross-validation (Training metrics)
partitionedModel =
crossval(classificationNeuralNetwork, 'KFold', 5);
validationPredictions = kfoldPredict(partitionedModel);
train_accuracy = (1 - kfoldLoss(partitionedModel)) *
100;
% Testing metrics
testPredictions = predict(classificationNeuralNetwork,
test_data(:, predictorNames));
test_accuracy = sum(testPredictions == test_data.Target)
/ numel(test_data.Target) * 100;
% Calculate costs
costMatrix = ones(4) - eye(4); % 0 diagonal, 1 elsewhere
confMat_train = confusionmat(response,
validationPredictions);
confMat_test = confusionmat(test_data.Target,
testPredictions);
train_cost = sum(sum(confMat_train .* costMatrix));
test_cost = sum(sum(confMat_test .* costMatrix));
% Calculate metrics
[train_recall, train_precision, train_f1] =
calculate_metrics(confMat_train);
[test_recall, test_precision, test_f1] =
calculate_metrics(confMat_test);
% Create results tables
resultsTable = table(round(train_accuracy,1), train_cost,
round(test_accuracy,1), test_cost, ...
'VariableNames', {'Train_Acc', 'Train_Cost',
'Test_Acc', 'Test_Cost'}, 'RowNames', {'NN'});
metricsTable = table(round(train_recall,3),
round(train_precision,3), round(train_f1,3), ...
round(test_recall,3), round(test_precision,3),
round(test_f1,3), ...
'VariableNames', {'Train_Rec', 'Train_Pre',
'Train_F1', 'Test_Rec', 'Test_Pre', 'Test_F1'}, ...
'RowNames', {'NN'});
end
function [recall, precision, f1] =
calculate_metrics(confMat)
% Calculate macro-average metrics from confusion
matrix

```

```

n = size(confMat,1);
recall_vec = zeros(n,1);
precision_vec = zeros(n,1);
for i = 1:n
    recall_vec(i) = confMat(i,i) / sum(confMat(i,:));
    precision_vec(i) = confMat(i,i) / sum(confMat(:,i));
end

```

Appendix C. Bayesian optimization framework for the ONN

This appendix describes the methodology and hyperparameter search space used for the Bayesian optimization of the ONN. The process was carried out to automatically identify the most effective neural architecture and training parameters for the bearing fault diagnosis task, going beyond manually predefined model structures.

C.1 Optimization setup

Objective	Minimize 5-fold cross-validation classification error
Algorithm	Bayesian optimization with Gaussian process
Iterations	30 iterations
Implementation	MATLAB Classification Learner toolbox

C.2 Bayesian optimization search space configuration

Hyperparameter	Search Space	Description
Number of Layers	1, 2, 3	Number of fully connected layers
First Layer Size	[1, 300]	Number of neurons in first hidden layer
Second Layer Size	[1, 300]	Number of neurons in second hidden layer
Third Layer Size	[1, 300]	Number of neurons in third hidden layer
Activation Function	ReLU, Tanh,	Activation Function
Regularization Strength (Lambda)	[1e-5/n, 1e5/n]	Regularization Strength (Lambda)
Standardize Data	True, False	Standardize input predictors before training