Leveraging Lightweight AI for Anomaly Detection in Mechanical power transmission at the Edge

Walid Ayadi[1,a], Amar Amouri[1,b*], Saeed Ali[1,c], Nada Alzahmi[1,d] and Mohammad Nadeem Khalid[1, e]

[1]Abu Dhabi Polytechnic, Institute of Applied Technology, Abu Dhabi, UAE

[a]ayadi.walid@gmail.com, [b*]amar.amouri@actvet.ac.ae, [c]A00060712@actvet.ac.ae, [d]A00057047@actvet.ac.ae, [e]er.nkhalid@gmail.com

Keywords: Anomaly Detection, TinyML, Neural Networks, Predictive Maintenance, power transmission.

Abstract

The integration of IoT and TinyML platforms has revolutionized fault detection and condition monitoring in electromechanical systems, enabling real-time data acquisition and analysis in resource-constrained environments. This study presents an implementation of TinyML model for mechanical power transmission fault detection on DC electric motors. The proposed model is deployed on Arduino Nano 33 BLE Sense equipped with an Inertial Measurement Unit (IMU). A 3-axis fusion sensor IMU is used to get vibration signals including acceleration, angular rate and orientation. An in -house developed data set was used to check the performance of the proposed AI model. A meticulously designed dataset comprising eleven operation conditions, encompassing idle, noisy, loose, and misaligned states, was developed. Time-domain and frequency-domain features were extracted and used to train a Multi-Layer Perceptron (MLP) model, achieving an overall classification accuracy of 97.1%. The model's deployment on the Arduino edge demonstrated real-time fault detection capabilities with 73.19% accuracy on unseen data, leveraging minimal computational resources. The results showcase the potential of TinyML-based systems for predictive maintenance, providing a cost-effective and scalable solution for industrial applications.

1.Introduction

The rapid advancement of systems has underscored the need for robust and efficient techniques for fault detection in electromechanical systems. Among these, DC electric motors play a critical role in various industrial applications due to their simplicity, efficiency, and controllability. However, their performance can be adversely affected by various faults, including mechanical misalignments, loose assembly components, and abnormal operational states. Identifying these faults in real-time is essential to ensure efficient mechanical power transmission, minimize downtime, and reduce maintenance costs [1]. The advancements in embedded systems have enabled the development of compact, cost-effective solutions for condition monitoring of rotating machinery [2]. In particular, the Arduino Nano 33 BLE Sense, equipped with an inertial measurement unit (IMU), has emerged as a versatile platform for capturing high-fidelity vibration data and running machine learning models [3]. The present study leverages this platform to collect and analyze multi-dimensional data, including accelerometer, gyroscope, and magnetometer readings, to classify various motor states under diverse operating conditions. Using a combination of time and frequency-domain feature extraction techniques [4,5], a robust dataset is developed for training a neural network-based classification model. This approach enables effective anomaly detection while optimizing resource utilization, making it suitable for TinyML applications [6,7,8]. The growing interest in predictive maintenance has driven extensive research in the field of fault detection for electromechanical systems [9,10,11,12]. Conventional methods often relied on heuristic analysis and signal processing techniques, such as time-domain and frequency-domain analysis, to detect and classify motor faults. For instance, Fourier Transform and Wavelet Transform have been widely used to extract frequency-specific features indicative of specific fault conditions [13,14]. While these techniques provide valuable insights, their effectiveness is often limited by the complexity of real-world signals and the absence of advanced classification mechanisms. Recent advancements in machine learning have revolutionized the fault detection paradigm by enabling automated feature extraction and classification. Neural networks, especially Multi-Layer Perceptron (MLP) model and Convolutional Neural Networks (CNNs), have been employed to achieve high accuracy in identifying fault conditions [15]. For example, X. Li et al. [10] demonstrated the use of deep learning techniques for identifying bearing faults in rotating machinery, achieving significant improvements over traditional methods. Similarly, Y. Zhang et al. [11] highlighted the effectiveness of neural networks in detecting misalignment and unbalance in motors using vibration data. The integration of IoT and TinyML platforms has further enhanced the accessibility and scalability of condition monitoring systems [12]. Platforms such as the Arduino Nano 33 BLE Sense enable real-time data acquisition and processing, making them ideal for on-site implementation. Papaioannou, A et al. [16] showcased the potential of low-power embedded systems in predictive maintenance, emphasizing the importance of optimizing resource utilization without compromising classification accuracy. This paper presents an implementation of TinyML methodology for fault detection in power transmission on a DC electric motor, utilizing an in-house developed dataset collected using a custom-built module based on the Arduino Nano 33 BLE Sense. The proposed framework demonstrates high classification accuracy with minimal computational overhead, paving the way for efficient real-time implementations in industrial settings. The present study addresses this gap by creating a meticulously designed dataset encompassing eleven distinct operations states: Ideal, noisy, loose, and misaligned conditions. By combining this dataset with a carefully constructed feature extraction and classification framework, the study aims to provide a robust solution for fault detection in DC electric motors. Furthermore, this work aligns with the emerging focus on lightweight machine learning models tailored for resource-constrained environments

2. Method

A. Data collection and hardware setup:

In this paper, an in-house developed dataset was collected using an Arduino Nano 33 BLE and a Lab setup of an electromagnet DC motor coupled to a balanced load with bearing. This board is considered as ultra-low power embedded devices able to run ML with low-cost resources (TinyML). It has a 32-bit ARM Cortex-M4F microcontroller running at 64 MHz with 1 MB of ROM and 256 KB of RAM. The hardware board is fixed on the top end of the motor where the X axis is aligned and oriented towards the shaft of the motor Sense as shown in Figure 1. This is an arbitrary positioning but will be used as a reference for the real-time configuration. A built-in RGB Led is used as visual indicator of the predicted categories (Normal, Misalignment, Loose).

Figure 1. The experimental setup showing the DC electric Motor coupled to the load and the Arduino board on the top end of the motor

The hardware module collected vibrations of the DC electric motor using its built-in LSM9DS1 IMU (Inertial Measurement Unit) fusion sensor including a gyroscope, accelerometer and magnetometer sensors. The raw data is collected in the cartesian (x, y, z) coordinates for the three sensors as shown in Table 1. Each row of data contains the timestamp in microseconds with 9 values at 32 bits. The sampling rate is 100 Hz. This leads to having 900 values every 1 second. Since the data is time dependent, the window frame of a single sample is defined by 1000 ms with a window sliding of 1000 ms (no overlapping). This choice is based on the visual inspection of the cyclic motion at the lowest speed of 540 rpm (9Hz) where the cyclic period is T=111 ms which is enough to capture the pattern within the window frame. Figure 2 shows the plot raw data of the sample of a loose status at 540 rpm. Some samples have been removed from the data collection when extra noise has been noticed due to experimentation issues. A total of 753 samples of clean dataset are retained where 80% is used for training and 20% for validation balanced between all classes. An additional 120 samples are used for testing.

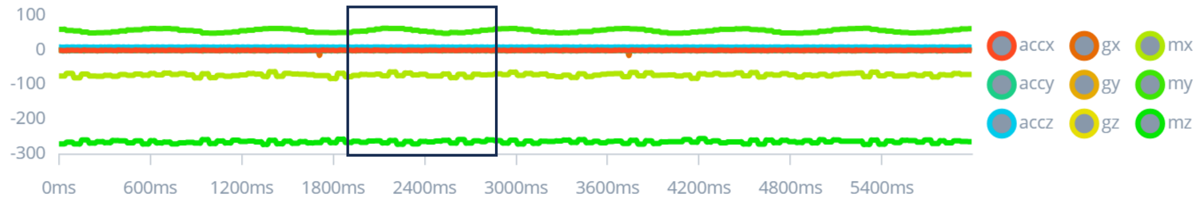

Figure 2. Raw data plot of the Loose sample at 540 rpm with a window frame of 1000 ms. x-axis and y-axis represent the time in ms and the collected raw value.

The datasets cover 11 classes based on the motor speed and the mechanical power condition which provides the source of fault. The classes are categorized into 4 main Types: Ideal, Normal, Loose and Misalignment under the same load with three different speeds: 540, 1050 and 1168 rpm as shown in table 2.

Table 2 Dataset details for electric motor conditions including: type, applied voltage, motor speed.

No.	Class	Type	Speed [RPM]
1	Class 1	Idle	0
2	Class 2	Idle- Noisy	0
3	Class 3	Normal	540
4	Class 4	Normal	1050
5	Class 5	Normal	1168
6	Class 6	Loose	540
7	Class 7	Loose	1050
8	Class 8	Loose	1168
9	Class 9	Misalignment	540
10	Class 10	Misalignment	1050
11	Class 11	Misalignment	1168

Data collection was conducted by running the motor through various operational states to capture as many conditions as possible. In the Idle state, the motor is at a standstill and not rotating; this state was used as the reference state for a standstill. The Idle-Noisy state involved external disturbances such as natural external disturbance while the motor was in an idle state. The misalignment state is created by an artificial angular misalignment affected in the coupler between the motor and the load. The angular misalignment is measured by a dial gauge as amplitude of vibration which draws 0.240 mm per 100 mm measured in length from the coupling center. This value is selected beyond the accepted tolerance of 0.1mm per 100mm as per the manufacturer. The Loose state is created by loose bolts, which was indicative of mechanical instability in the mechanical power transmission. This status generates a compound type of mechanical misalignment and other vibration modes which is considered as a special class in our work. This is to detect the source of anomaly whether it is generated from pure angular misalignment or from loose assembly. These states have been carefully designed to present different scenarios for anomaly detection to acquire and analyze robust data.

B. Data Preprocessing.

The preprocessing step first keeps all raw data, which was collected from an experiment test in lab for different motor speed and power transmission conditions. The raw data has been collected from the Arduino board using an SD card in separate text files. Each experiment records more than 60s of time series data with a sampling rate of 100Hz. These files have been exported to a CSV file with labelled data. Every row in the CSV file consists of 10 values: timestamp and 9 sensors data: a_x, a_y, a_z representing accelerometer; m_x, m_y, m_z representing magnetometer; and g_x, g_y, g_z representing gyroscope readings. The head and tail of the files have been visually investigated and trimmed to maintain data quality and integrity ensuring a minimum of 60 s of clean data. The accelerometer data range is set at [-4, +4]g -/+0.122 mg, The gyroscope range is set at [-2000, +2000] dps +/-70 mdps, and the magnetometer range is set at [-400, +400] uT +/-0.014 uT.

C. Feature Extraction and neural network architecture

Feature extraction transformed the dataset into a more informative representation for classification. Spectral features have been used to extract frequency and power using a digital signal processing algorithm. Time-domain features such as Root Mean Square (RMS), skewness, and kurtosis were calculated to quantify vibration signal amplitude and distribution. Using Fast Fourier Transform (FFT), the raw time-domain data was converted into the frequency domain with 1s window and no overlap. Frequency-domain features included spectral power calculations across critical frequency bands (e.g., 3.12–9.38 Hz and 15.62–21.88 Hz), which were identified as significant for distinguishing motor states. This comprehensive process generated 117 features in the frequency domain, capturing essential patterns for anomaly detection. The combination of time and frequency-domain analysis allowed for a detailed representation of the motor's behavior, capturing both the signal characteristics and their distribution across critical frequency bands.

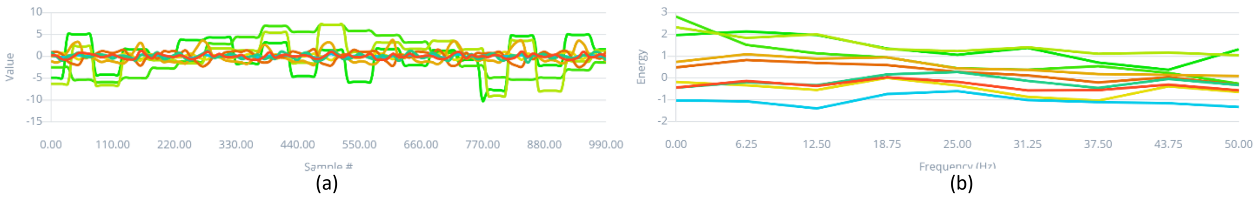

Figure 3. DSP signal (a), and Spectral log (b) of the processed signal from a window frame in the Loose sample at 540 rpm

The machine learning model is based on a Multi-layer Perceptron (MLP), which is a standard deep neural network. The deployed model specifications are shown in table 3. This model was trained on the 117 features generated previously. The results which will be presented in the next section are based on the model with all 13 extracted features from each sample of 1s.

Table.3 Machine learning model specification

Layer	Details
Input Layer	117 represents the generated features
Hidden Layer 1	Dense layer with 58 neurons and ReLU activation
Hidden Layer 2	Dense layer with 22 neurons and ReLU activation
Output layer	Dense layer with SoftMax activation for 11 classes
Optimizer	Adam optimizer
Learning Rate	0.0005
Batch size	32

3. Results and discussion

This section presents the classification results using the most significant features. The confusion matrix for the 11 classes is shown in Table 1. The accuracy is evaluated by the percentage of correct predictions out of all predictions. The precision is measured by the percentage of true positive predictions out of all positive predictions made by the model; therefore, the recall is the percentage of actual positive instances that the model correctly identified. The F1 Score represented as the harmonic mean of precision and recall, needed to balance the trade-offs between false positives and false negatives predictions. The following equations have been used for the calculation:

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

$$Precison = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precison + Recall}$$

Where TP, TN, FP and FN are the True Positive, True Negative, False Positive and False Negative predictions. The confusion matrix has been converted into a one-vs-all classes matrix (binary-class confusion matrix) for calculating class-wise metrics. The results in **Error! Reference source not found.** present the performance regarding Precision, Recall, F1 score, and the area under the ROC curve of the model.

Table 1. Normalized confusion matrix of the tested model. (Values are given in percentage)

		Predicted class										
		Idle		**Normal**			**Loose**			**Misalignment**		
		1	2	3	4	5	6	7	8	9	10	11
True class	1	**100**	0	0	0	0	0	0	0	0	0	0
	2	0	**100**	0	0	0	0	0	0	0	0	0
	3	0	0	**88.9**	0	0	0	0	11.1	0	0	0
	4	0	0	0	**100**	0	0	0	0	0	0	0
	5	0	0	0	25	**75**	0	0	0	0	0	0
	6	0	0	0	0	0	**100**	0	0	0	0	0
	7	0	0	0	0	0	0	**100**	0	0	0	0
	8	0	0	0	0	0	0	0	**100**	0	0	0
	9	0	0	0	0	0	0	0	0	**92.9**	7.1	0
	10	0	0	0	0	0	0	0	0	0	**100**	0
	11	0	0	0	0	0	0	0	0	0	0	**100**

Table 2. Evaluation metrics for model performance

Metric	Value
Accuracy	97.1%
Weighted F1 Score	0.97
Precision	0.97
Recall	0.97
Area Under ROC Curve	1.00

The developed model is deployed on Arduino Nano 33 BLE sense using TensorFlow Lite with no optimization (float 32bits). The processing time required to obtain the data and extract features is 59 ms and 4.6KB of RAM. The inference time is 2ms using 1.9KB of RAM and 44.3KB of ROM for prediction as shown in Table 3. The test of unseen data run on the edge shows an accuracy of 73.19% with a threshold of 70% in prediction trust. The Arduino board has demonstrated a good capability to run the model in real time with lightweight resources.

Table 3. On-device resources and performance

	Spectral Features	Classifier	Total	Allocated Resources
Latency	59ms.	2ms.	**61ms**.	100ms
Ram	4.6K	1.9K	**4.6K**	256KB
Flash	-	44.3K	**44.3K**	1024KB
Accuracy			73.19%	

Overall, the performance analysis of this model is very good, with overall accuracy at 97.1%, weighted F1 score, precision, and recall of 0.97 meaning that its classification quality and balance are quite good across the dataset. Table 4 shows that most classes are well identified, although there is minor misclassification mainly among those closely related classes. For instance, Class 9 is 7.1% misclassified but still predicted within the same category of misalignment. Class 5 has an overlap of 25% with Class 4 but remains in the same category as Normal. There is also a misclassification of Class 3 with Class 8 between Normal and Loose respectively. An AUC at 100% shows that for all possible thresholds, this model has always ranked the true classes higher than the false one. This represents great overall separability of the classes, although for specific misclassifications, at a fixed threshold, some might be observed in the confusion matrix. The accuracy of 73.19% in real time processing on device is acceptable for this application compared to the low resources utilized where only 61% of latency, 1.8% of RAM, and 4.3% of ROM are utilized. The device can detect the fault after a reasonable time after many attempts with repeatability. This result can be improved for more reliability by training the model with more datasets under extended operational mode. The extracted features can be adjusted for more separability focusing on the significant parameters. An adequate filter may eliminate noisy features which are ignored in this work. The device has remaining resources which can be used later to add electrical data from the motor to enhance the model.

4. Conclusion

This study demonstrates the feasibility and effectiveness of leveraging IoT-enabled TinyML platforms for fault detection in DC electric motors. By utilizing an Arduino Nano 33 BLE Sense, high-fidelity vibration data was collected and analyzed using a combination of time-domain and frequency-domain feature extraction techniques. The resulting dataset, encompassing eleven distinct motor states, enabled the development of a robust MLP-based classification model. The model achieved an impressive classification accuracy of 97.1% during validation and 73.19% during on-device testing, highlighting the potential for real-time implementation in industrial environments. Despite minor misclassifications observed in closely related motor states, the system's overall performance, resource efficiency, and scalability make it an excellent candidate for predictive maintenance applications. Future work will focus on expanding the dataset to include additional operational scenarios, optimizing feature extraction for enhanced separability, and integrating electrical data to improve fault detection accuracy. By addressing these areas, the proposed framework can further improve reliability, providing industries with a cost-effective, lightweight solution for maintaining the operational efficiency of electromechanical systems.

References

[1] S. B. Lee et al., Condition Monitoring of Industrial Electric Machines: State of the Art and Future Challenges, IEEE Industrial Electronics Magazine, vol. 14, no. 4, pp. 158-167, Dec. 2020

[2] Omar AlShorman, et al., Advancements in condition monitoring and fault diagnosis of rotating machinery: A comprehensive review of image-based intelligent techniques for induction motors, Engineering Applications of Artificial Intelligence, Volume 130, 2024

[3] Sai Charan, K., An Auto-Encoder Based TinyML Approach for Real-Time Anomaly Detection, *SAE Int. J. Adv. & Curr. Prac. in Mobility* 5(4):1496-1501, 2023

[4] E. Njor, M. A. Hasanpour, J. Madsen and X. Fafoutis, "A Holistic Review of the TinyML Stack for Predictive Maintenance," in *IEEE Access*, vol. 12, pp. 184861-184882, 2024

[5] Rodrigue Tambeck Nguimfack, et Al., Domain adaptation between heterogeneous time series data: A case study on real-time rotary machinery fault diagnosis, Manufacturing Letters, pp 1535-1543, Volume 41, Supplement, 2024

[6] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki and A. S. Hafid, "A Comprehensive Survey on TinyML," in *IEEE Access*, vol. 11, pp. 96892-96922, 2023

[7] P. Verma and H. Gupta, "A review on lightweight neural networks for embedded systems," Embedded Systems Design Journal, vol. 14, pp. 25-36, Jan. 2023

[8] W. Ayadi, A. Saidi and I. Channoufi, "Exploring Human Activity Patterns: Investigating Feature Extraction Techniques for Improved Recognition with ANN," 2024 IEEE 7th International Conference on Advanced Technologies, Signal and Image Processing (ATSIP), pp. 188-193, 2024

[9] A. Garcia, M. Clark, and N. White, "Implementation of IoT sensors in predictive maintenance," Sensors Journal, vol. 21, pp. 345-360, Apr. 2023

[10] X. Li, J. Ding, and Y. Chen, "Deep learning-based bearing fault diagnosis for rotating machinery," IEEE Transactions on Industrial Electronics, vol. 67, no. 12, pp. 10989-10998, Dec. 2020.

[11] Y. Zhang, K. Wang, and S. Liu, "Fault detection in motors using neural networks and vibration data," Mechanical Systems and Signal Processing, vol. 152, pp. 107368, Jan. 2021.

[12] M. Patel, S. Kumar, and P. Singh, "IoT and TinyML in predictive maintenance: A case study on resource-constrained systems," IEEE Internet of Things Journal, vol. 9, no. 4, pp. 2337-2346, Apr. 2022.

[13] J. Smith and A. Brown, "Wavelet transform-based fault diagnosis for electrical motors," International Journal of Advanced Manufacturing Technology, vol. 112, pp. 3456-3465, Nov. 2020.

[14] R. Davis et al., "Time-frequency analysis for anomaly detection in industrial motors," IEEE Access, vol. 11, pp. 789-800, Feb. 2023.

[15] Ceylan, N., Sönmez, E. & Kaçar, S. Cost effective detection of uneven mounting fault in rotary wing drone motors with a CNN based method. *SIViP* 18, pp 8049–8059, 2024

[16] Papaioannou, et Al., D. LP-OPTIMA: A Framework for Prescriptive Maintenance and Optimization of IoT Resources for Low-Power Embedded Systems. *Sensors*, vol *24*, 2125, 2024