Physics-informed Intelligent Motor Fault Detection

Sinian Li*, Raj Thilak Rajan*, Edmund Marth†, Patrick Zorn†, Wolfgang Gruber† and Justin Dauwels*
**Signal Processing Systems, Delft University of Technology, Delft, The Netherlands*
† Institute for Electric Drives and Power Electronics, Johannes Kepler University Linz, Linz, Austria

Abstract—**Intelligent Fault Detection (IFD) has garnered significant attention, with recent advances in AI-empowered predictive maintenance. A key challenge in applying IFD models lies in the interpretability of the methods, since the mechanisms are typically complex and difficult to integrate with data-driven approaches. In addition, the integration of edge devices is an emerging trend, which ensures fault detection and subsequent decision making on the edge, and thus offering an instant response as compared to a conventional centralized server-based architecture. However, to realize Edge-based IFD the primary constraints are low storage capacity and limited computational resources. In this paper, we address various critical challenges in automatic Edge-based IFD for motors in industrial settings, focusing on three key constraints, i.e., (a) limited availability of training data, (b) the lack of method interpretability, and (c) the computational and storage limitations of edge devices. To overcome these challenges, we propose a suite of light weight Physics-Informed (PI) AI algorithms to achieve Edge-based IFD - without compromising detection performance. We validate our proposed methods on experimental data for motor fault detection, and additionally present results from the implementation of these methods on an edge device. We discuss the benefits of our proposed solutions, and give directions for future work.**

Index Terms—**Fault detection, Condition monitoring, Electric motor, Machine learning, Edge AI**

I. Introduction

Intelligent Fault Detection (IFD) has recently gained increased attention due to the limitations of manual maintenance, where a delay in defect identification can lead to significant system failures and unexpected operational costs. Hence, IFD techniques are widely explored for industrial machines and systems failures, such as in the field of automotive manufacturing, wind turbine generators, robotics and power infrastructure [1]. IoT applications, including unmanned fault detection, typically rely on edge devices that are capable of analyzing data and making decisions locally. Thanks to the growing capabilities, declining costs, and miniaturization of electronic hardware, the deployment of IFD models on low-end microcontrollers is now commercially feasible [2]. Embedded computing platforms, such as FPGA [3], Raspberry [4] and Arduino [5] are commercially available to be applied in scale under industrial scenarios for a variety of challenges [6].

Unlike model-based fault detection approaches, which rely on mathematical models of system behavior [7], data-driven approaches use training data to capture relationships between inputs and outputs, and deep models for high-level feature representation [1]. End-to-end deep models directly learn relationships from the raw data, and achieve high detection accuracy by bypassing intermediate feature engineering steps and discovering best features needed during training [8]. However, this high accuracy and flexibility comes at the cost of computational efficiency. In the scope of Edge-based IFD, deep models are not the first choice due to the limited computational resources and storage capacity.

In contrast, shallow classification models are relatively lightweight, but require significantly more pre-processing of collected data. Feature engineering is a crucial pre-processing step to reduce the overall computational complexity and improve the accuracy of detection [9]. Digital signal processing (DSP) including Fourier Transform (FT) [10], Short-Time Fourier Transform (STFT) [11], and Signal Imaging [12] play an important role in time-frequency domain analysis and general feature extraction. Transforming data into different domains reveals hidden characteristics that may be obscured in the time domain. With the engineered features, shallow machine learning algorithms can be applied to classify faults. For example, Support Vector Machines (SVMs) are widely used in fault detection and diagnosis due to their effectiveness in handling high-dimensional data and complex decision boundaries [13]. As an example, Edge2Train is a novel framework designed to enable microcontrollers to train SVM models locally with real implementation experiments [14]. Another recent work detects open-circuit faults and remains the safety of three-phase Pulse-width Modulation (PWM) rectifier using the Random Forests (RF) algorithm [3]. Comparative evaluation shows that RF tends to outperforms SVM and CNNs methods, and is typically employed on a FPGA controller [3]. A simplified CNN is designed to mitigate overfitting by integrating variations in motor speed [15]. For simplicity and efficiency, Extreme Learning Machine (ELM) is used for classifying faults with features extracted from vibration signals (CWRU) [9].

To address the trade-off of accuracy and efficiency, physical knowledge of the system can be integrated into the framework. In [16], the vibration signals are demodulated via the Hilbert transform, and their spectra are used to select characteristic sub-bands for bearing faults. These sub-bands serve as engineered features for classification. The authors in [17] propose a Physics-Informed CNN (PICNN), which integrates a novel feature weighting layer into the CNN. The bearing fault characteristic frequencies are embedded through the constraints of the distribution of attention parameters.

The authors would like to thank R-PODID - Reliable Powerdown For Industrial Drives for providing essential resources and support. This work was supported by the Chips Joint Undertaking and its members under Grant Agreement n° 101112338.

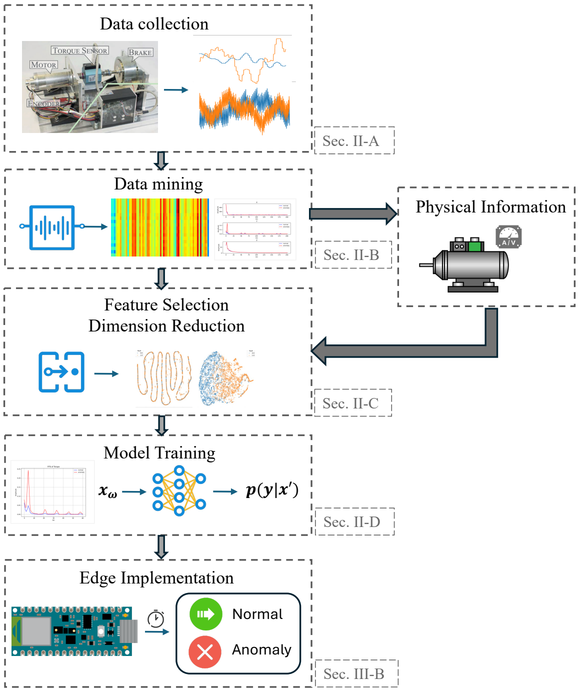


Fig. 1: Our proposed Physics-informed data processing and model deployment pipeline

Although existing Physics-Informed (PI) methods address bearing faults [16], [17], several challenges and opportunities remain. Current approaches mainly focus on vibration signals, yet torque and current signatures also provide valuable insights. Most existing research targets bearing faults, leaving other issues less explored. This work specifically addresses commutation angle errors, including the exploration for edge application. Future research can extend to other fault types and broader applications.

In this paper, we aim to present a low-cost Physics-Informed (PI) data processing framework for Edge-based IFD, with a focus on motor commutation angle error detection. Our objectives are as follows:

- We present frequency-domain insights of motor commutation angle error through data mining. We further leverage the known electrical frequency and sidebands of the motor to guide PI feature selection, which enhances the interpretability of the proposed IFD framework.
- Second, we propose a suite of motor fault detection methods, which are adaptations of diverse Machine Learning models for resource-constrained hardware, and we provide a detailed evaluation of the proposed methods for motor fault identification.
- Third, to validate the proposed method and demonstrate its effectiveness, we design a test bench experiment to collect a dataset from a motor operating under both nominal and faulty working conditions. We then deploy the trained models on a microcontroller board for real-time fault detection evaluation.

The rest of the paper is organized as follows. Section II introduces experimental design, proposed feature extraction and engineering, and classification models. Section III presents results and analyses. Section IV concludes this paper.

II. Methodologies

In this work, we aim to provide a solution for lightweight motor fault detection without sacrificing the detection performance. Figure 1 shows the overall workflow of our proposed framework, which consists of data collection, data mining, physics-informed feature selection, model training and hardware implementation for real-world deployment. In this section, we discuss (a) test bench data collection, (b) preprocessing techniques, (c) integration of characteristic frequencies in feature engineering, and (d) multiple classifiers.

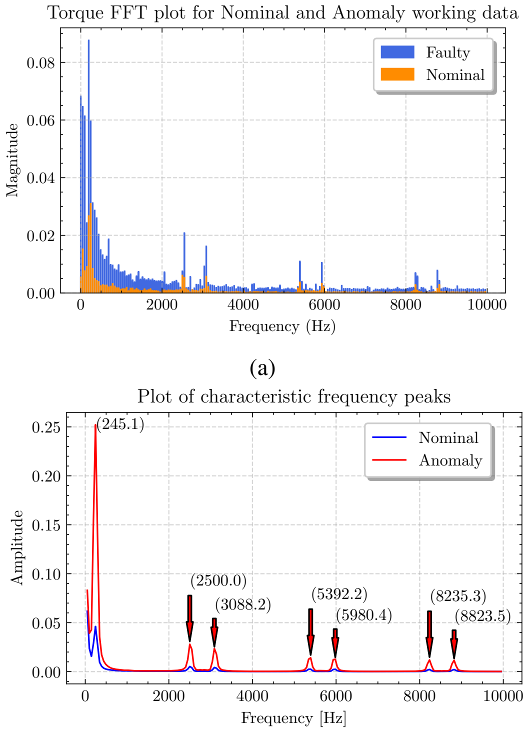
A. Experimental design

Our test bench consists of a *ECI 63.40 K4* brushless servomotor from EBM-Papst, which is equipped with sensors to collect current, torque, and rotational speed signals at a sampling rate f_s of 20 kHz. The motor has 4 pole pairs (p_z), and operates at a speed of $n = 4000$ rpm to deliver a torque $T = 670$ mNm. A commutation angle error is typically caused by misalignment of the Hall sensors leading to a rotational offset of φ_Δ (in electrical degrees, $^{\circ_{el}2}$). A failure occurs if the offset between the actual and expected sensor positions deviates from $30^{\circ_{el}2}$. This error typically results in a semi-optimal operating point, which can lead to abnormal behaviors such as increased torque ripples, unstable phase currents, and elevated electrical stress. There are a equal number of $M = 2000$ measurements from both nominal and faulty working conditions, making the total dataset size 4000. For both nominal and faulty conditions, each type of measurement contains $2L = 408$ samples, where L denotes half of the number of samples in each measurement.

B. Preprocessing & feature extraction

For M measurements collected under nominal condition, Discrete Fourier Transform (DFT) is applied to each of them, and the positive frequencies of resulting complex signals are collected into columns of matrix $\mathbf{X}_\mathrm{N} \in \mathbb{C}^{L\times M}$, where N denotes the nominal working condition. Then, for the DFT result of each measurement, only the magnitude spectra is preserved, which gives matrix $|\mathbf{X}_\mathrm{N}| \in \mathbb{R}^{L\times M}$. Finally, the average over M magnitude spectra is calculated and collected in vector $\mathbf{x}_\mathrm{N} \in \mathbb{R}^{L}$. Similarly, the positive-frequency DFT results of measurements under faulty condition and their magnitude spectra can be collected into matrix $\mathbf{X}_\mathrm{F} \in \mathbb{C}^{L\times M}$ and $|\mathbf{X}_\mathrm{F}| \in \mathbb{R}^{L\times M}$, respectively. Then, the average over the magnitude spectra is $\mathbf{x}_\mathrm{F} \in \mathbb{R}^{L}$. In Figure 2a, examples of the magnitude spectra for measurements from nominal and faulty working condition, respectively, are visualized. The salient peaks around the electrical frequency, i.e., $f_e = \frac{n\cdot p_z}{60}$, and side-peak frequencies can be observed.

Figure 2b annotates the peaks in the averaged signal spectra over the transformed measurements under nominal and faulty working conditions. The first peak is the motor electrical



(a)

(b)

Fig. 2: Plots of (a) frequency-domain torque signals and (b) averaged signal spectra under nominal and faulty working conditions

frequency, which in our case, is $f_e = \frac{n \cdot p_z}{60} = 266.67$ Hz. Since the experimental rotational speed is not always at 4000 RPM, the corresponding frequency is not exactly 266.67 Hz as expected. The other peaks are caused by the sample-and-hold operation. Observe the sidebands around the center frequency: $f_c = \frac{f_s}{\alpha}$, where f_s is the sampling frequency and α is the number of samples per operation, which is 7 in our case. The side peaks can be expressed as $f_p = kf_c \pm f_0, k = 1, 2, 3$, where f_0 is the fundamental frequency of input torque signals. These peaks remain consistent across measurements.

C. Physics-informed feature engineering

Given the frequency-domain representation of the signals and key frequency bins analyzed in II-B, the feature engineering process targets on selecting the most informative features for fault detection. The proposed feature engineering algorithm is outlined in Algorithm 1. The algorithm takes the averaged magnitude spectra $\mathbf{x}_\mathrm{N}$ and $\mathbf{x}_\mathrm{F}$ to locate the characteristic frequency bins. The z-scores $\mathbf{z}_\mathrm{N} \in \mathbb{R}^L$ and $\mathbf{z}_\mathrm{F} \in \mathbb{R}^L$ of $\mathbf{x}_\mathrm{N}$ and $\mathbf{x}_\mathrm{F}$ are then computed for each frequency index, after which we can obtain a vector $\mathbf{d} \in \mathbb{R}^L$ collecting their differences. Features are selected based on the difference exceeding a certain predefined threshold τ, which is adapted based on the data and the need for reducing computational complexity. The algorithm finally outputs the selected feature indices set, which is denoted by $\mathcal{S}$.

```
Require: Averaged magnitude spectrum for nominal $\mathbf{x}_\mathrm{N}$ and
    faulty $\mathbf{x}_\mathrm{F}$ working conditions, threshold $\tau$, electrical
    frequency $f_e$ and sidebands $f_p = kf_c \pm f_0, k = 1, 2, 3$
Ensure: Feature indices $\mathcal{S} = \emptyset$
  Identify key frequency bins: Define set $\mathcal{B}$ of frequency
    bins around $f_e$ and expected sidebands $f_p$
  for each $i \in \mathcal{B}$ do
      Compute the z-score $\mathbf{z}_\mathrm{N}[i]$ and $\mathbf{z}_\mathrm{F}[i]$ of $\mathbf{x}_\mathrm{N}[i]$ and
        $\mathbf{x}_\mathrm{F}[i]$ respectively, using class-specific statistics (mean
        $\mu_i$ and standard deviation $\sigma_i$)
      Compute the difference $\mathbf{d}[i] = |\mathbf{z}_\mathrm{F}[i] - \mathbf{z}_\mathrm{N}[i]|, \forall i \in \mathcal{B}$
  end for
  Selected feature indices: $\mathcal{S} \leftarrow \{i \in \mathcal{B} : \mathbf{d}[i] \geq \tau\}$
  return $\mathcal{S}$
```

Algorithm 1: Feature selection for motor fault identification

Figure 3 shows the t-sne visualization of the original samples, frequency-domain features, PI features, and MLP processed PI-features [18] under both nominal and faulty working conditions. Although spectrum-based analysis is widely used for vibration signals in motor fault detection, it remains under-explored for torque-based motor IFD. As seen here, the frequency-domain features exhibit a significant overlap between the two classes. In contrast, the proposed PI feature selection identifies the most relevant frequency components, resulting in a far clearer class separation and thus demonstrating the effectiveness of the proposed feature selection technique. Additionally, by reducing feature dimensionality, this approach saves both memory and computational resources. Moreover, the last plot of MLP processed PI-features shows that a fully connected neural network can further refine PI features to learn more robust patterns for fault detection.

D. Classification models

Given PI feature indices set $\mathcal{S}$, only the rows indexed are selected, as matrix $\mathbf{X}_\mathcal{S}$. Denoting i-th feature vector as $\mathbf{X}_{\mathcal{S},i}$, the training dataset of size m for this binary classification problem is written as $\mathcal{D} = \{(\mathbf{X}_{\mathcal{S},i}, y_i)\}_{i=1}^{m}, \mathbf{X}_{\mathcal{S},i} \in \mathbb{R}^{|\mathcal{S}|}, y_i = f(\mathbf{X}_{\mathcal{S},i}) \in \{0, 1\}$, where m is the total number of training samples after splitting the dataset and $f(\cdot)$ is the labeling function that maps each feature vector $\mathbf{X}_{\mathcal{S},i}$ to a binary label. In inference phase, trained models map the input feature $\mathbf{X}_{\mathcal{S},i}$ to discrete class label $\hat{y}_i$, predicting the condition of the motor. In this work, we have multiple practical constraints, i.e., low-dimensional features for highly constrained on-device memory and computation, and real-time prediction requirement. Under these conditions, shallow ML models such as Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB), Support Vector Machine (SVM) and a Multiple Layer Perceptron (MLP) model are well-suited [18]. Specifically, a compact MLP strikes a balance between capturing high-level patterns from low-dimensional data and avoiding heavy memory or computational overhead of a deep architecture. These methods are briefly summarized below.

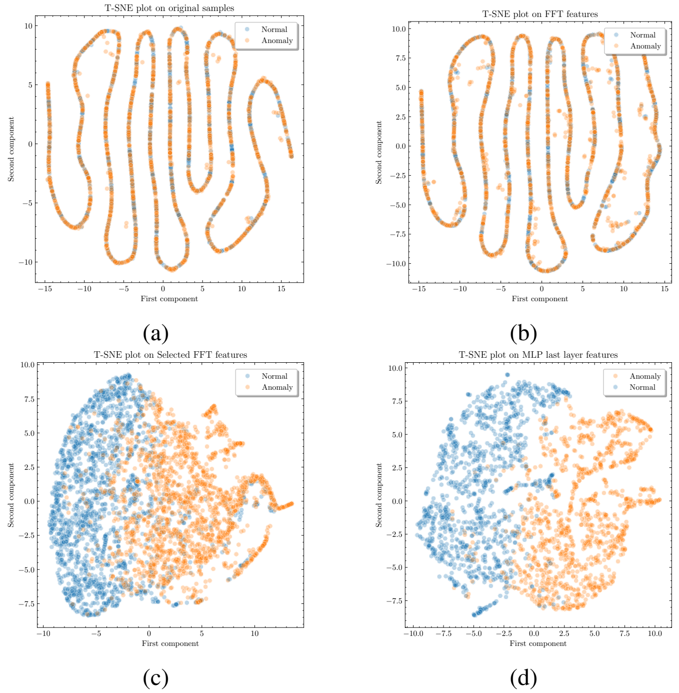


Fig. 3: T-sne visualizations of (a) original samples, (b) original frequency-domain features, (c) proposed PI features and (d) MLP processed features under both nominal and faulty working conditions

Decision Tree (DT): DTs recursively split the input space to partition data into classes. Given the training set $\mathcal{D}$, the splitting structure is usually decided by information gain, which is defined as $\text{IG}(\mathcal{D},\mathcal{F}) = \text{Entropy}(\mathcal{D}) - \sum_{v\in\text{Values}(\mathcal{F})} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} \cdot \text{Entropy}(\mathcal{D}_v)$, where $\mathcal{F}$ is the feature set, $\mathcal{D}_v$ is the subset of $\mathcal{D}$ that has value v from feature set $\mathcal{F}$, and $\text{Entropy}(\mathcal{D}) = -\sum_{c\in\text{Classes}} p(c)\log p(c)$ is the entropy of the dataset $\mathcal{D}$.

Random Forest (RF): RF extends decision trees by constructing an ensemble of n_T trees, each trained on a random subset of the training data. Given an input vector $\mathbf{x} \in \mathbf{X}_{\mathcal{S}}$, the t-th tree produces a prediction $h_t(\mathbf{x})$. The final prediction is the majority vote of the individual trees: $\hat{y} = \text{mode}\left(\{h_t(\mathbf{x})\}_{t=1}^{n_T}\right)$.

Gradient Boosting (GB): GB model is trained in a stage-wise manner. The model at stage $m+1$ is updated as $F_{m+1}(\mathbf{x}) = F_m(\mathbf{x}) + \eta h_m(\mathbf{x})$, which is a combination of the current model $F_m(\mathbf{x})$ and the newly fitted tree $h_m(\mathbf{x})$ that approximates the residual error of the loss function with η to be the learning rate.

Support Vector Machine (SVM): SVM classifier finds the hyperplane that best separates classes in feature space. For a linear SVM, the hyperplane is found by solving the optimization problem: $\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$ subject to $y(\mathbf{w}^T\mathbf{x}+b) \geq 1, \forall\mathbf{x} \in \mathbf{X}_{\mathcal{S}}$, where $\mathbf{w}$ is the weight vector and b is the bias.

Multiple Layer Perceptron (MLP): For an MLP with one hidden layer, given a feature vector $\mathbf{x} \in \mathbf{X}_{\mathcal{S}}$, the output of the hidden layer is computed as $\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x}+\mathbf{b}_1)$, where $\mathbf{W}_1$ is the weight matrix, $\mathbf{b}_1$ is the bias vector, and σ is the activation function, e.g., ReLu or sigmoid. The output of the network is computed via the softmax function as $\hat{y} = \text{softmax}(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2)$. The parameters are learned through back-propagation by minimizing a loss function, e.g., cross-entropy loss.

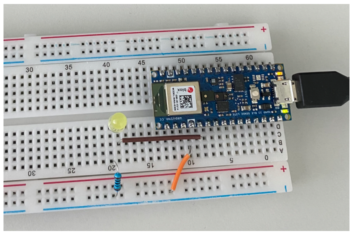


Fig. 4: Picture of our Arduino board in a circuit

III. Experiments

In this section, we introduce the edge device specifications, on which we migrate trained models to conduct real-time fault detection. We apply the proposed PI feature engineering solution on the test-bench data for evaluation.

Hardware specifications: The hardware used is Arduino Nano 33 BLE Sense. The board is powered by a 64 MHz Arm Cortex-M4F processor, with 1 MB of flash memory and 256 kB of RAM. The hardware implementation is carried out on a circuit with an LED notification, as shown in Figure 4.

Evaluation Metrics: Accuracy is defined as $\text{Accuracy} = \frac{1}{2M}\sum_{i=1}^{2M} \mathbf{1}\left(y_i = \hat{y}_i\right)$, where $2M$ is the number of total measurements in both nominal and faulty working condition, and $\mathbf{1}$ represents indicator function. Additional metrics include the number of model parameters, memory usage, and the inference time. These metrics provide insights into the model's complexity, resource requirements, and operational efficiency.

A. Evaluation on collected dataset

The performance of different feature engineering methods is reported in Table I. The input dimensions are: **Current signals** - 408 (baseline), 204 (frequency-domain), 33 (statistical); **Torque signals** 408 (baseline), 204 (frequency-domain), 15 (engineered). This directly influences the number of parameters and therefore the storage size.

Our proposed PI features, combined with an MLP classification model, achieve the highest test accuracy of 97.8%. Notably, the PI features require less than half the dimensionality of conventional statistical features while improving accuracy by 12.7%. Compared to frequency-domain features, our approach enhances accuracy by 4.5% while it reduces feature dimensionality by 86.4%, significantly minimizing storage requirements. To further validate the effectiveness of PI features, we validate them across multiple lightweight classification models. Remarkably, four out of five models exhibit a significant accuracy improvement when incorporating PI features, demonstrating their robustness and generalizability.

B. Edge implementation and evaluation

Table II summarizes the performance of the implemented models on the arduino board as shown in Figure 4. After

TABLE I: Accuracy performance of different models with current signals and torque signals with/without feature engineering

Model	Current signals			Torque signals		
	Baseline (Raw)	**Frequency**	**Statistical**	**Baseline (Raw)**	**Frequency**	**Proposed**
DT	68.8	51.2	77.9	61.5	92.3	93.9
RF	77.0	55.3	86.0	64.4	94.9	95.4
GB	74.4	51.4	81.6	54.8	96.0	95.1
SVM	60.5	53.3	83.2	49.5	94.3	94.8
MLP	85.0	55.7	93.6	45.1	97.0	97.8

TABLE II: Performance of models with PI features

Model	# Parameters	Memory [kB]	Time [ms]	Accuracy
DT	233	97	0.48	93.9
RF	22328	412	1.20	95.4
GB	1472	291	2.89	95.1
SVM	12082	399	36.54	94.8
MLP	1601	102	0.95	97.8

applying our proposed feature engineering, these lightweight models can be deployed on this edge device with limited memory and computational capacity. In industrial settings, real-time requirements and latency constraints often demand efficient inference on such devices. The MLP model achieves the highest accuracy with the second-lowest memory consumption and inference time, highlighting the potential of tiny ML in IFD even under stringent memory and delay constraints. Remarkably, using the same MLP structure with the baseline input size of 408, yields over 40k model parameters, which is about 25 times of the model trained on the PI-based feature set. Meanwhile, the statistical feature set produces the second smallest model at 2.9k parameters, yet this is still 1.81 times larger than the PI-based model. These reductions demonstrate the effectiveness of the proposed PI features in reducing storage overhead while preserving the detection accuracy, making them well-suited for edge AI applications.

IV. Conclusion

In this work, we proposed a novel framework of combining physics-based feature extraction with efficient Machine Learning models for Edge-based IFD, and validated using hardware implementations using test bench data. Our experiments show that the proposed PI feature selection technique reduces the feature dimensionality without compromising detection accuracy, and thus alleviates the storage and computational overhead. One of the many directions for future work is to investigate the effect and choice of various hyperparameters e.g., choice of τ, on the performance of the proposed methods.

References

[1] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, 2020.

[2] S. Lu, J. Lu, K. An, X. Wang, and Q. He, "Edge computing on iot for machine signal processing and fault diagnosis: A review," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11 093–11 116, 2023.

[3] L. Kou, C. Liu, G. wei Cai, Z. Zhang, J. ning Zhou, and X. mei Wang, "Fault diagnosis for three-phase pwm rectifier based on deep feedforward network with transient synthetic features," *ISA Transactions*, vol. 101, pp. 399–407, 2020.

[4] D. Park, S. Kim, Y. An, and J.-Y. Jung, "Lired: A light-weight real-time fault detection system for edge computing using lstm recurrent neural networks," *Sensors*, vol. 18, no. 7, p. 2110, 2018.

[5] S. U. Jan, Y.-D. Lee, J. Shin, and I. Koo, "Sensor fault classification based on support vector machine and statistical time-domain features," *IEEE Access*, vol. 5, pp. 8682–8690, 2017.

[6] H. Ren, D. Anicic, and T. A. Runkler, "Tinyol: Tinyml with online-learning on microcontrollers," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[7] R. Isermann, "Model-based fault-detection and diagnosis-status and applications," *Annual Reviews in control*, vol. 29, no. 1, pp. 71–85, 2005.

[8] D. T. Hoang and H. J. Kang, "A motor current signal-based bearing fault diagnosis using deep learning and information fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3325–3333, 2020.

[9] N. Sikder, A. S. Mohammad Arif, M. M. Islam, and A.-A. Nahid, "Induction motor bearing fault classification using extreme learning machine based on power features," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8475–8491, 2021.

[10] N. Sikder, K. Bhakta, A. Al Nahid, and M. M. M. Islam, "Fault diagnosis of motor bearing using ensemble learning algorithm with fft-based preprocessing," in *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, 2019, pp. 564–569.

[11] A. H. Boudinar, A. F. Aimer, M. E. A. Khodja, and N. Benouzza, "Induction motor's bearing fault diagnosis using an improved short time fourier transform," in *Advanced Control Engineering Methods in Electrical Engineering Systems*. Cham: Springer International Publishing, 2019, pp. 411–426.

[12] G. Xu, M. Liu, Z. Jiang, W. Shen, and C. Huang, "Online fault diagnosis method based on transfer convolutional neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 509–520, 2020.

[13] M. Fei, L. Ning, M. Huiyu, P. Yi, S. Haoyuan, and Z. Jianyong, "On-line fault diagnosis model for locomotive traction inverter based on wavelet transform and support vector machine," *Microelectronics Reliability*, vol. 88-90, pp. 1274–1280, 2018, 29th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis (ESREF 2018).

[14] B. Sudharsan, J. G. Breslin, and M. I. Ali, "Edge2train: a framework to train machine learning models (svms) on resource-constrained iot edge devices," in *Proceedings of the 10th International Conference on the Internet of Things*, ser. IoT '20. New York, NY, USA: Association for Computing Machinery, 2020.

[15] J.-H. Han, D.-J. Choi, S.-K. Hong, and H.-S. Kim, "Motor fault diagnosis using cnn based deep learning algorithm considering motor rotating speed," in *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, 2019, pp. 440–445.

[16] S. Shen, H. Lu, M. Sadoughi, C. Hu, V. Nemani, A. Thelen, K. Webster, M. Darr, J. Sidon, and S. Kenny, "A physics-informed deep learning approach for bearing fault detection," *Engineering Applications of Artificial Intelligence*, vol. 103, p. 104295, 2021.

[17] H. Lu, V. Pavan Nemani, V. Barzegar, C. Allen, C. Hu, S. Laflamme, S. Sarkar, and A. T. Zimmerman, "A physics-informed feature weighting method for bearing fault diagnostics," *Mechanical Systems and Signal Processing*, vol. 191, p. 110171, 2023.

[18] M. Kubat, *An Introduction to Machine Learning*, 3rd ed. Springer Cham, 2021, vol. 4-6.