IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 November 2023, accepted 17 December 2023, date of publication 18 December 2023, date of current version 9 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3344676

RESEARCH ARTICLE

Advanced Dual RNN Architecture for Electrical Motor Fault Classification

YOUSEF ALKHANAFSEH[1], TAHIR CETIN AKINCI[1,2], (Senior Member, IEEE), EMINE AYAZ[1], AND ALFREDO A. MARTINEZ-MORALES[3]

[1]Electrical Engineering Department, Istanbul Technical University, 34469 Istanbul, Turkey
[2]WCGEC, University of California, Riverside, Riverside, CA 92521, USA
[3]Bourns College of Engineering, Center for Environmental Research and Technology, University of California, Riverside, Riverside, CA 92521, USA

Corresponding author: Tahir Cetin Akinci (tahircetin.akinci@ucr.edu)

ABSTRACT In recent years, there has been a remarkable increase in the usage of Deep Neural Networks (DNNs) for addressing and solving electrical field problems. This research primarily aims to present an advanced approach to classify different motor faults based on their time-series data by implementing a new Recurrent Neural Network (RNN) model that consists of mixed Long short-term memory (LSTM), Gated Recurrent Unit (GRU), and two Fully Connected (FC) layers. The main idea of this study centers on developing one comprehensive model capable of categorizing primary motor faults. The proposed model is supposed to classify 10 different classes, extracted from the Machinery Fault Database (MaFaulDa), which are normal (no-fault), vertical misalignment, horizontal misalignment, imbalance, overhang-ball, overhang-cage, overhang-outer race, underhang-ball, underhang-outer race, and underhang-cage. Classifying 10 different situations can be considered as a notable classification problem. Additionally, the learning period did not include any data augmentation, which reflects the model's power in training over the available data. Significantly, the accuracy of the model is enhanced by setting precise values for hyperparameters, including network structure (number of layers and neurons), learning rate, regularization, optimizer type, number of epochs, and more. The obtained train-validation-test accuracies from the proposed model are 99.87%, 99.599%, and 99.48%, respectively. The accuracy of the model represents the highest accuracy among other publications. This advanced approach offers numerous advantages, including early-stage fault detection, improved robustness in industrial maintenance, and generating fast and intelligent alerts, thereby reducing the possible damage to electrical instruments.

INDEX TERMS Condition monitoring, GRU, LSTM, motor faults classification, recurrent neural networks.

I. INTRODUCTION

In the early 1960s, significant developments in the global industrial landscape have begun, drawing considerable attention to electrical motors. Consequently, several different types of motors were invented, such as the AC synchronous inductor motor, brushless DC motor, and stepper motor [1]. With the ongoing development of electrical motors, their structures have become more efficient and complex [2], [3]. Therefore, their maintenance is considered an essential concern. Currently, electrical motors play a pivotal role in the industry and can be found in almost all types of devices [4]. They possess desirable features such as high stability, durability, and ease of use and maintenance, which have contributed to their widespread usage. However, under certain conditions, such as low insulation resistance, vibrations, overheating, etc., which can negatively impact electrical motors, they may suffer from faults or even become worthless if left unprotected [5], [6], [7]. As a result, it is crucial to detect faults in electrical motors as early as possible to protect workers, prevent substantial economic losses in manufacturing facilities, and reduce potential damage that could disrupt operations [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Ines Domingues.

Currently, the protection and maintenance of electrical motors are considered major functions that must be carefully implemented [1].

Initially, straightforward fault detection techniques were employed, such as over-current or over-voltage detection, which primarily involved measuring different parameters such as current and voltage and comparing them with predefined threshold values. If the measured values exceed these thresholds, the affected motor is shut down. However, disabling affected motors may not be acceptable in specific circumstances, rendering these methods inadequate and ineffective [9]. Currently, motor fault detection techniques primarily fall into five main categories: model-based, signal-based, knowledge-based, hybrid-based, and active methods [10]. These techniques are applied after extracting the necessary information from the analyzed source. This information typically includes one of two main feature types: domain features such as time-frequency characteristics or computation features such as aggregated statistics [11].

Over the past few years, there has been a significant increase in interest in analyzing data from diverse sources, leading to numerous research endeavors related to Machine Learning (ML), Artificial Neural Networks (ANN), and Recurrent Neural Networks (RNNs) focused on using these analyses to address electrical machine problems [12]. Accordingly, this research fully implements a specially designed RNN model. The heart of our model comprises of Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and two Fully Connected (FC) layers with varying numbers of neurons, resulting in a highly intricate yet robust architecture. Furthermore, we place a strong emphasis on hyperparameter tuning, carefully optimizing critical elements such as network architecture (number of layers and neurons), learning rate, regularization techniques, choice of optimizer, number of training epochs, and more. These meticulous adjustments contribute significantly to the model's outstanding performance. Our proposed model sets out to categorize ten distinctive fault classes, meticulously extracted from the Machinery Fault Database (MaFaulDa) [13]. These classes encompass a spectrum of scenarios, ranging from normal motor operation to specific fault conditions, including vertical misalignment, horizontal misalignment, imbalance, and various overhang and underhang faults. This broad classification scope represents a substantial challenge that our model is uniquely equipped to address. Both the included scenarios and the obtained accuracy are the highest among published studies with ten classes and train-validation-test accuracies of 99.87%, 99.599%, and 99.48%, respectively. It is important to note that the validation step is done using 5-fold cross-validation. Moreover, an ablation study is conducted on the proposed model to understand the importance of its components individually. The reason for creating a model containing GRU, FC, and LSTM layers rather than traditional machine learning algorithms, such as the one proposed in [14], in conducting this classification task can be illustrated in three main points, which are better at capturing the temporal dependencies of sequential data, the ability to learn and extract complex features from sequential data automatically, and higher flexibility in dealing with different input sizes, and the ability to increase the model complexity so that it can extract non-linear relationships. Apart from its academic relevance, our study brings forth numerous benefits. It opens the door to early detection of faults in electrical machinery, strengthens the resilience of industrial maintenance, enables the generation of intelligent alerts, and increases the potential to significantly decrease harm to electrical equipment,

The organization of the paper is as follows: after the introduction, the previous works on the same subject are investigated in detail in the Related Work section. Thereafter, the models and mathematical equations used are briefly described in the Background section. Consequently, the main architecture and the structure of the proposed model are meticulously explained in the System Architecture section. Later, the steps taken while applying the proposed model are sequentially detailed in the Implementation Details section. Last but not least, the obtained results are presented, and finally, a comprehensive summary is provided in the Conclusion section.

II. RELATED WORK

Commonly, the literature emphasizes that using the time domain data is not efficient for determining the important features of faults [15]. As a result, the frequency domain or time-frequency domain is usually employed during data preprocessing. In the field of motor fault diagnosis, several studies have been conducted to classify faults using the data from the MaFaulDa database. These studies vary in their approaches, including the use of traditional ML methods such as the Support Vector Machine (SVM) algorithm, similarity-based modeling (SBM), and both artificial neural networks (ANN) and RNNs.

The authors of [6] conducted a comprehensive exploration of SBM for fault detection, with a particular focus on novel approaches to model training and similarity metrics. The research is illustrated through the identification of rotating machinery faults, where SBM serves as either a standalone classifier or a feature generator for a random forest classifier. They achieved an accuracy of 98.5% on the MaFaulDa database but did not provide training and validation charts, leaving the training procedure of the proposed model unclear. Additionally, they classified only 7 classes.

In the paper by [16], they introduced the Predictive Maintenance (PdM) model with Convolutional Neural Network (CNN), designed to classify faults in rotating equipment and provide guidance on when maintenance interventions should be initiated. Their results include two values: one related to the model's performance when trained over classes separately and another when trained on the classes together, achieving an accuracy of 97.3%. This research classifies the same number of faults as our study.

Another study that utilizes SBM for classifying fault identification in rotating machinery is presented in [17]. The research explores a novel training approach involving prototype selection. Experimental findings, including the MaFaulDa and other databases, demonstrate a weaker model than [6], with accuracies of 96.4% and 98.7% on the MaFaulDa database. It also classifies fewer faults, specifically 5.

A new approach for a multi-fault classification system for electric rotating machinery leveraging an ANN and Synthetic Minority Over-sampling (SMOTE) is introduced in [18]. Notably, the model incorporates the relief feature selection method to enhance performance by identifying influential features. It utilizes the MaFaulDa database as its data source and achieves an accuracy of 97% while classifying 6 situations.

The study of [19] discusses the Hamiltonian neural networks (HNN) approach, which extends beyond conventional black-box models by incorporating physical constraints governing mechanical systems. The proposed HNN is trained on observational data to capture the system's conserved energy in normal and abnormal scenarios. They used the obtained weights vector to distinguish 6 faults taken from the MaFaulDa database, achieving accuracies of 78% for binary classification (normal vs. abnormal) and 84% for the multi-class problem (normal and five distinct abnormal faults).

The study in [20] introduces a multi-fault classification system employing ANN for automated differentiation of faults in rotating machinery. Key features include rotation frequency and statistical metrics such as mean, entropy, and kurtosis. The model's efficacy is increased through the utilization of SMOTE to address class imbalance and identify influential features. The proposed method is applied to the MaFaulDa database and achieves an accuracy of 97.1%. This method is tested over 6 classes.

This study [21] introduces an Early Classification (EC) approach for Structural Fault Recognition (SRF) that balances fault prediction accuracy and earliness. It works based on two sequential steps: a deep learning classifier prioritizes accuracy, and an early decision policy is applied. It uses LSTM and GRU separately in its architecture and depends on two metrics: accuracy and earliness. The achieved accuracy is 98.32%, and the obtained earliness is 55.68% on the MaFaulDa database. However, this research considers only 4 classes: normal, imbalance, horizontal misalignment, and vertical misalignment.

A dual decision-making strategy is discussed in [22], utilizing a CNN with fuzzy recurrence plot (FRP) for one stream and an LSTM network for distinctive frequency components (DFC). Additionally, it introduces a DFC-based ranking and image combining scheme to select key sensor signals and generate 3D-FRP and 2D flattened FRP (F-FRP) representations. When the model is tested on the MaFaulDa database, it achieves up to 99.0% accuracy. However, like the study in [21], only 4 classes are classified.

T4PdM is presented in [23], introducing a modified Transformer model for automatic fault classification in PdM. This model achieves exceptional accuracy of 99.98% on the MaFaulDa database and 98% on the CWRU database. However, this study excludes the overhang-bearing data completely, making its accuracy incomparable with other studies. Furthermore, their model made two mistakes; predicted normal situation as horizontal misalignment and cage fault as normal. On the other hand, our proposed model uses all samples found in the database and it only makes one mistake as it predicts one horizontal misalignment fault as a vertical misalignment fault.

III. BACKGROUND

In this section, the primary background of the techniques utilized in this research is intended to be explained. It includes three main subsections: Neural Network (NN), Metrics, and Signal Processing. Within each subsection, a brief explanation and the mathematical equations associated with each method are discussed.

A. NEURAL NETWORK

This research primarily leverages LSTM and GRU layers and the Sparse Categorical Cross-Entropy loss function, which are elaborated upon in this section.

LSTM is a type of RNN that is first introduced to the memory cells and gating mechanisms in RNN [24], [25]. There are two main variables in LSTM which are cell state (c_t) and hidden state (h_t) which are responsible for capturing information from the sequence units. LSTM model calculates the current value of the c_t using Equation 1, and the value h_t using Equation 3. It is obvious from Equation 1 that candidate cell state ($\tilde{c}_t$) must be previously calculated which can be briefly explained as the information that is assumed to be added to the c_t, see Equation 2.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{1}$$

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2}$$

$$h_t = o_t \odot g(c_t) \tag{3}$$

Its architecture mainly contains three different control gates which are input (i_t), forget (f_t), and output (o_t) [26]. The equations of these gates are displayed below in Equation 4, Equation 5, and Equation 6, respectively.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

GRU can be considered as a type of RNN architecture that is used to make sequential data processing tasks. It is first introduced to be able to capture features/dependencies of the recurrent units based on different occurrences [27]. In short, GRU is known for its efficiency and ability to capture long-range dependencies in sequences. The internal structure of GRU has two gates: the reset gate (r_t) and the update gate

(z_t) [28]. The first one decides whether to forget or to keep the information from the previous state, see Equation 8. The other gate is responsible for controlling the mixing process between the previous and new states, see Equation 9. It is clear that the sigmoid function is used in both Equation 8 and Equation 9 as it adds the capability to gates to whether keep or forget the information as it gives a value between 0 and 1, see Equation 7. There are two more important concepts which are hidden state (h_t) and candidate activation ($\tilde{h}_t$). The hidden state can be thought of as the memory that can encode information from previous units, see Equation 10. GRU uses it to make predictions on the current unit. On the other hand, candidate activation is the information that is used by GRU to update the current (h_t), see Equation 11.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{7}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{8}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{9}$$

$$h_t = (1 - z_t) h_{t-1} + z_t \tilde{h}_t \tag{10}$$

$$\tilde{h}_t = tanh(W x_t + U(r_t \odot h_{t-1})) \tag{11}$$

In the previous equations, 2 and from 8 to 11, x_t is the input at time step t, W and U are the trainable parameters (weights), h_{t-1} is the previous hidden state, $\odot$ is an element-wise multiplication, and *tanh* is the hyperbolic tangent function which is used as the activation function for GRU. In general, LSTM and GRU models can be used as classifiers, as they are used in the current research, and as forecasters as they are used in other research [29].

Sparse Categorical Cross entropy is a commonly used loss function in ML, particularly in deep learning and NN training. It is primarily used for classification tasks when the target values are provided as integers representing class labels. Its working principle is built on computing the cross-entropy for each data point by comparing the predicted class probabilities to the true class label. Its mathematical formula is represented in Equation 12 below.

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{e^{\hat{y}_i}}{\sum_{j=1}^{C} e^{\hat{y}_j}} \right) \tag{12}$$

where $L(y, \hat{y})$ represents the loss value, N is the number of samples or data points, C is the number of classes, i iterates over the samples (from 1 to N), j iterates over the classes (from 1 to C), y_i represents the true class label (an integer) for the i-th sample, $\hat{y}_i$ represents the predicted probability distribution over classes for the i-th sample.

B. EVALUATION METRICS

In general, there are four main metrics used to evaluate ML classification models: accuracy, precision, recall, and F1 score [30]. Accuracy, as shown in Equation 13, is one of the most important metrics that represent the number of correct predictions (True Positives (TP) and True Negatives (TN)) divided by the total number of predictions (the sum of TP, TN, False Positives (FP), and False Negatives (FN)). Furthermore, precision is a metric that measures the number of TP of the target class made by the model, as seen in Equation 14. Another metric is Recall, which assesses the model's ability to find all the correct samples of the target class, as described in Equation 15. Sometimes, recall and precision alone may not provide accurate results. In such cases, the F1 score metric can be used, which is calculated by taking the harmonic mean of Precision and Recall [31], [32], as shown in Equation 16.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{13}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

C. SIGNAL PROCESSING

The arbitrary signal is a signal without a specific pattern, thereby no way to describe it [33]. On the other hand, the mean and variance, see Equations 17 and 18, respectively, of stationary signals do not change over time [34].

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{17}$$

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{18}$$

where μ is the mean, N is the number of values in the data, X_i is an individual value in the data, i is the iteration number over data ranges from 1 to N, and finally σ^2 is the variance

Another important concept is sampling frequency which refers to the rate at which the samples of a signal are captured. In addition, the Nyquist Theorem states that to be able to reconstruct a signal, the used sampling frequency must be at least twice the highest frequency component in the original signal.

The main purpose of Fourier Transform (FT) is to give frequency information from the time domain signal. Its equation is represented in equation 19. However, it does not provide this information with its existence time. Thus, it is nearly useful in stationary signals, signals whose values are constant over time.

$$F(w) = \int_{-\infty}^{\infty} f(t) \cdot e^{-iwt} dt \tag{19}$$

To pass this insufficiency, the Short-Time Fourier Transform (STFT) is developed with the intention that the time-frequency representation is provided. Its main process is done by assuming some parts of the signal as stationary parts, splitting them implementing the window function, and then adding them up by calculating their FT, see Equation 20.

$$F(\tau, w) = \int_{-\infty}^{\infty} f(t) \cdot w(t - \tau) \cdot e^{-iwt} dt \tag{20}$$

The window function of STFT is marked with $w(t-\tau)$, translation/time localization parameter is used as τ. The only difference between the equations of FT and STFT is that in STFT there is one multiply part which is the window function.

Another type of Fourier transform is the Fast Fourier Transform (FFT). During this research, FFT is utilized for stationary signals. Unlike FT which is used for continuous signals, FFT is utilized for discrete signals, refer to Equation 21. Another advantage related to optimization enhancement is that computation reduction can be obtained using FFT as well.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j(2\pi/N)kn} \tag{21}$$

Here, $X(k)$ represents the k-th frequency component in the frequency domain, $x(n)$ is the input signal in the time domain, N is the number of samples in the input signal, and j is the imaginary unit.

The only filter used while preparing the data is the Wiener filter. It can be briefly considered as a filter that is used to effectively enhance signals and images by reducing their noises [35], [36]. The main equation of the Wiener filter is shown below in Equation 22

$$H(f) = \frac{S_{xy}(f)}{S_{xx}(f)} \tag{22}$$

where $H(f)$ is the frequency domain representation of the Wiener filter, $S_{xy}(f)$ is the cross-power spectral density between the desired signal and the observed signal, and $S_{xx}(f)$ is the power spectral density of the observed signal.

As a normalization method, z-score technique is used. Its equation is shown in Equation 23. This method, in general, plays an important role in increasing the convergence rate of the NN and decreasing the needed number of epochs to complete the train process [37].

$$z = \frac{x-\mu}{\sigma} \tag{23}$$

where z is the z-score (standard score) of the input value, x is the original input data point, μ is the mean of the data, and σ is the standard deviation of the data.

IV. SYSTEM ARCHITECTURE

The preprocessing stages and the proposed methodology of this project are prominently illustrated in Figure 1. The architecture begins at the top with electrical instruments and an electrical motor. The motor continuously generates ten different time-series data while it is running. These data are then collected and loaded into a database, specifically the MaFaulDa database. Starting from this point, the preprocessing stages commence by verifying whether these data sequences are stationary or not. Based on the stationary check, the spectrum of the stationary data is obtained using the FFT, while the spectrum of the non-stationary data is acquired using the STFT. Subsequently, both results from these transformations pass through a new preprocessing stage: sampling. This step is taken because the spectrum of the data contains a significant dead band that lacks important information, it is shown in the Implementation details section. Sampling reduces the data's range, thereby enhancing the training process. Next, a Wiener filter is applied to reduce noise in the data which improves the quality of the data. Following this, the data are normalized using z-score normalization, which scales all the data to have a zero mean and a standard deviation of 1. This indirectly impacts the model training process, enhancing model convergence and improving generalization. After normalizing the data, it is split into two main parts: training and validation, and test, with ratios of 90% and 10%, respectively. As a result, the model was trained and validated on 1747 samples, and finally, it was tested on 194 samples. The training data are then used to teach the model the specific patterns of each fault. Moreover, the model is validated using 5-fold cross-validation with 80% and 20% of data for training and validation steps, respectively. Additionally, the model is fed validation data to monitor and assess the training process. The final data, test data, is employed to evaluate the model's performance when presented with new data. If the model's result indicates no fault, the process loops back to the database, and the same procedure is repeated; otherwise, the detected fault triggers an alarm.

The proposed model's structure, see Figure 2, is inspired by the Thyristor structure, which combines a four-layer semiconductor device, consisting of alternating P-type and N-type materials (PNPN) [38]. It primarily consists of two RNN layers: GRU and LSTM, with a Fully Connected (FC) or dense layer between them and a final output FC layer. The first layer is the GRU layer, comprising 512 neurons with a tanh activation function. The choice of GRU as the first layer, rather than LSTM, is based on its simpler structure, efficiency in mitigating over-fitting, and ability to capture short-term dependencies. The second layer of our model is FC with 128 neurons and ReLU activation function. This layer is pivotal in our classification problem, responsible for feature aggregation, non-linearity through ReLU activation, and improving class-specific patterns by encoding relationships between extracted features and classes. The result from this FC layer is then fed to the LSTM layer, which contains half the number of neurons compared to the GRU layer but uses the same activation function. LSTM helps handle long-term dependencies and deal with data noise. Lasso regression regularization (L1) with a 0.0025 value is applied within this layer to avoid over-fitting. The final layer is an FC layer with ten neurons, corresponding to the number of classes the model is expected to predict. Each neuron provides a probability related to its input class, and the highest probability determines the model's prediction. Thus, this layer facilitates the classification decision. An overview of our proposed model's internal structure is stated in Table 1. This table can be interpreted as follows: the first column, Layer (type) contains the name of each layer. In the output shape column, the first item of the tuple represents the batch

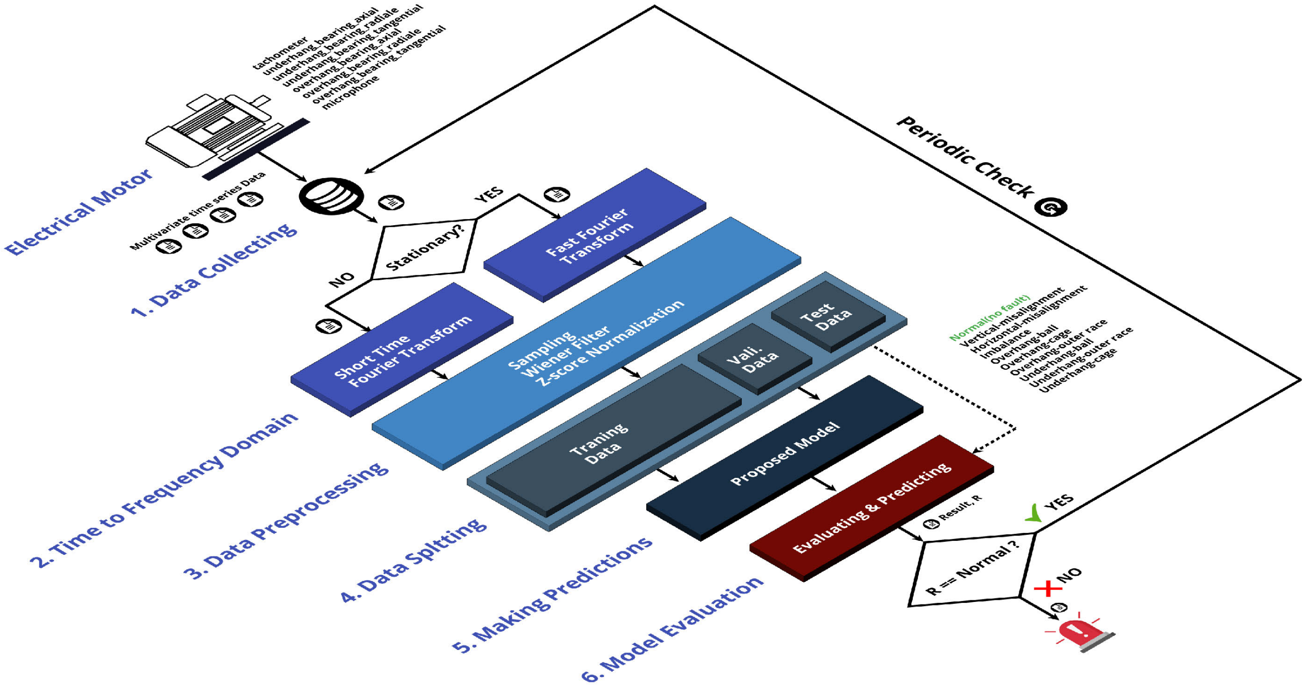

FIGURE 1. Project main architectural framework.

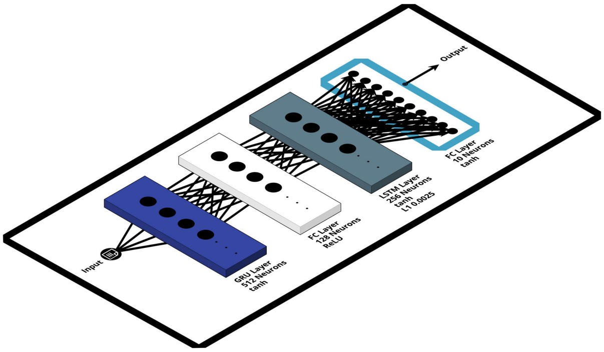

FIGURE 2. The internal structure of the proposed model.

TABLE 1. Proposed model structure overview.

Layer (type)	Output Shape	Parameters
gru (GRU)	(None, 1250, 512)	801792
dense (Dense)	(None, 1250, 128)	65664
lstm (LSTM)	(None, 256)	394240
dense_1 (Dense)	(None, 10)	2570

size, which is not explicitly given in the model summary as it depends on the training batch size, the second item of the tuple represents the length of the output sequence data, and the third one represents the number of neurons of that layer. Finally, the Parameters column shows the number of trainable parameters per layer.

The hyperparameters of the proposed model and their values are drawn in Table 2. They include learning rate, epochs, batch size, optimizer, GRU and LSTM activation functions, dense activation function, Dense_1 activation function, L1 regularization, loss function, and used metric. Subsequently, the proposed model comprises 1,264,266 trainable parameters, making it a relatively lightweight model.

TABLE 2. Key hyperparameters of the proposed model.

Hyperparameter	Value
Learning Rate	0.00065
Epochs	50
Batch Size	32
Optimizer	Adam
GRU Activation Function	tanh
LSTM Activation Function	tanh
Dense Activation Function	ReLU
Dense_1 Activation Function	tanh
Lasso Regression (L1)	0.0025
Loss Function	SparseCategoricalCrossentropy
Performance Metric	Accuracy

V. MaFaulDa DATABASE

This research is built on open-source data, taken from the MaFaulDa database. The system used to collect this data is prepared by first collecting vibration data using two different accelerometers in three different directions: axial, radial, and tangential. Moreover, there is a tachometer and a microphone equipped with the system to measure rotation frequency and capture operating sound, respectively. As a result, the data contains eight different features: a tachometer, underhang bearing axial, underhang bearing radial, underhang bearing tangential, overhang bearing axial, overhang bearing radial,

TABLE 3. Additional information about the MaFaulDa database.

Feature	Value
Sampling frequency	50 kHz
Time interval	5s
Motor power	1/4 CV DC
Frequency range	700-3600 rpm
Rotor speeds	10 - 60 Hz
System weight	22 kg
Axis diameter	1.6 mm
Axis length	52.0 mm
Rotor diameter	15.24 cm
Bearings distance	390 mm

TABLE 4. Number of samples per class.

Situation	Samples
Normal	49
Imbalance	333
Horizontal Misalignment	197
Vertical Misalignment	301
Bearing Cage Underhang	188
Bearing Outer Underhang	184
Bearing Ball Underhang	186
Bearing Cage Overhang	188
Bearing Outer Overhang	188
Bearing Ball Overhang	137

TABLE 5. Non-stationary signal samples.

Fault	Data File	Feature	Δ M.	Δ Var.
underhang/cage_fault	20g/60.6208.csv	M	0.14	18.2
underhang/cage_fault	20g/61.2352.csv	M	0.08	10.27
underhang/cage_fault	35g/56.1152.csv	M	72.7	4319.17
underhang/cage_fault	35g/32.5632.csv	M	0.06	11.19
underhang/cage_fault	35g/29.4912.csv	M	0.1	15.94
underhang/cage_fault	35g/56.5248.csv	M	0.13	10.32
imbalance	29.9008.csv	M.	0.08	13.76
overhang/ball_fault	35g/13.312.csv	UBA	0.04	11.31
overhang/ball_fault	6g/53.248.csv	UBT	0.03	12.09
overhang/ball_fault	0g/25.8048.csv	UBA	0.03	11.29

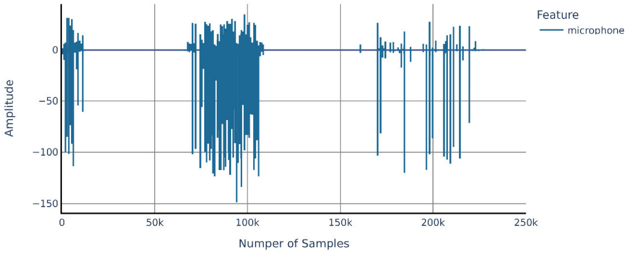

FIGURE 3. Sample of non-stationary data related to microphone feature obtained from (underhang/cage_fault/20g/60.6208.csv) data file.

overhang bearing tangential, and microphone. One sample from the MaFaulDa database is presented graphically in the following section. Each captured data is collected using a 50 kHz sampling rate over a 5-second time interval. Additional information related to the sampling rate, time interval, frequency range, motor power, rotor speeds, and diameter is stated in Table 3.

The MaFaulDa database mainly contains four different classes; the first one is normal operation and the other three are faults, unbalance, misalignments, and bearings. The last two faults have subcategories which are Horizontal and vertical for misalignments, and underhang and overhang for bearing faults as well. Leading to 6 overall categories. However, in this research, the cage, outer race, and ball faults of both overhang and underhang bearing faults are treated as separate categories. Therefore, this research is one of the few kinds of research that is implemented to detect 10 different situations related to MaFaulDa. The number of samples of each class is represented in Table 4, with 1951 total samples.

VI. IMPLEMENTATION DETAILS

In general, before starting any signal analysis, a thorough understanding of the given signals is paramount. Among the crucial aspects is grasping the signals' temporal nature, and how they change over time.

To determine whether the given signals exhibit stationarity or non-stationarity, the mean and variance approach is employed. By splitting each signal into 10 equal parts, 25k samples each, the mean and variance of each segment are independently calculated. If the difference between maximum and minimum variances is greater than 10, we consider this signal as the non-stationary signal. The calculations reveal that, except the stated signals in Table 5 below, all signals demonstrate stationarity. The columns of the table can be explained as follows; Fault is the Main Fault name, Data File is the file that contains that non-stationary feature, Feature represents the non-stationary feature, Δ M. is the result of max mean of the 10 different chunks minus the min mean of the 10 different chunks minus, and Δ Var. is the same as Δ M but for chunks' variances.

An example of the non-stationary signal is represented Figure 3. It is obvious from the figure that the signal has almost zero amplitude between 13K and 68K, 110K and 170K, and 230K and 250K, Nevertheless, the signal amplitude fluctuates in the intervals between 0 and 13k, 68K and 110K, and 170K and 230K. Therefore, when such a signal is passed through mean and variance stationary check, it will fail and it will be considered as a non-stationary signal.

The proposed processing steps, which precede the proposed model, mentioned in Figure 1, are subsequently represented in Figure 4 as separate figures. Only the workflow of the stationary signals is presented in this figure. The first Figure 4, (a) shows an example of the raw stationary data signals obtained from the MaFaulDa database. As mentioned before, each data file consists of eight distinct signals collected from T, OBA, OBR, OBT, UBA, UBR, UBT, and M, all listed at the bottom of the figure as legends. In addition, the data collected in MalFaulDa was generated at a 50 kHz sampling rate during 5 s, leading to a total 250.000 samples for each signal which is clearly exhibited at the x-axis of Figure 4 (a) with y-axis shows the voltage of these signals. These signals are obtained from the $vertical - misalignment/1.90mm/51.2.csv$ data file.

It is evident that Figure 4 (a) contains stationary signals as they have a specific pattern and do not change significantly over time. Since these are stationary signals, FFT is applied to the data, and the results can be seen in Figure 4 (b). The length of the transformed signals, the output of FFT or STFT, is the same as the input, resulting in 250k output points divided into 125k for positive and 125k for negative frequencies which is obvious from the x-axis of this figure. It is well known that when the input data consists entirely of real numbers (not imaginary), the obtained spectrum from FFT or STFT will be mirrored on both the positive and negative sides. Therefore, we decided to consider only the negative side, as shown in Figure 4 (c), which reduced the number of samples to 125k, thus improving the model's performance in terms of training time. Then, the x-axis is transformed into positive values, as altering the sign of the x-axis values has no impact on the underlying process. The obtained spectrum mostly exhibits high signal strength frequencies in two different spectrum bands: (0 - 25) kHz and (105 - 115) kHz, with a dead band at (25 - 105) kHz. As mentioned earlier, the raw data was generated using a 50 kHz sampling rate, resulting in a Nyquist frequency of 25 kHz. Therefore, the data's range of points can be reduced by re-sampling the data using a frequency lower than the Nyquist frequency. We decided to re-sample the data at a 100 Hz rate, resulting in 1250 samples for each sequence of data, as shown in Figure 4 (d). The x-axis of this figure shows the number of samples and the y-axis displays the range of their amplitudes. Based on this phase, the model training time is shortened as the number of samples decreases.

To improve the quality of these signals, we decided to reduce their noise using a Wiener filter. The Wiener filter is applied to the data sequences to enhance their quality by attenuating the amount of noise found inside these signals. It is known that this filter depends on two important parameters which are noise type and window size. The noise of this filter is estimated as the average of the local variance of the input, and the window size used to estimate the local statistics of the signal is set to 10. The result of this operation is shown in Figure 4 (e). It is evident from this figure that small noises are filtered, and the amplitudes of the signals are smoothed. Furthermore, the data are standardized using the z-score normalization model to enhance the convergence of the models, as shown in Figure 4 (f). This method is used to handle outliers and equalize the different data scales, making the data mean zero and their variance one. It is clear that the amplitude (y-axis) of each signal is normalized, changing the range of amplitudes from 0.332 and 6476.568 to -0.791 and 34.079. The normalization step is done separately on each signal found inside each data file. Table 6 illustrates the range of each column before and after the z-score normalization step. It is important to note that the before range refers to the range of the data obtained after applying the Wiener filter step. Based on that, it is clear from Figure 4 (f) that the microphone signal created a new peak exactly at 1100 Hz frequency. Actually, the microphone signal has a peak at this frequency, however, it is not clear in the previous figures as it is less than the other peaks. This occurred due to the z-score normalization of the microphone signal, which can be mathematically proven. For the microphone signal at a frequency of 1100, the value is 34.0796, with a

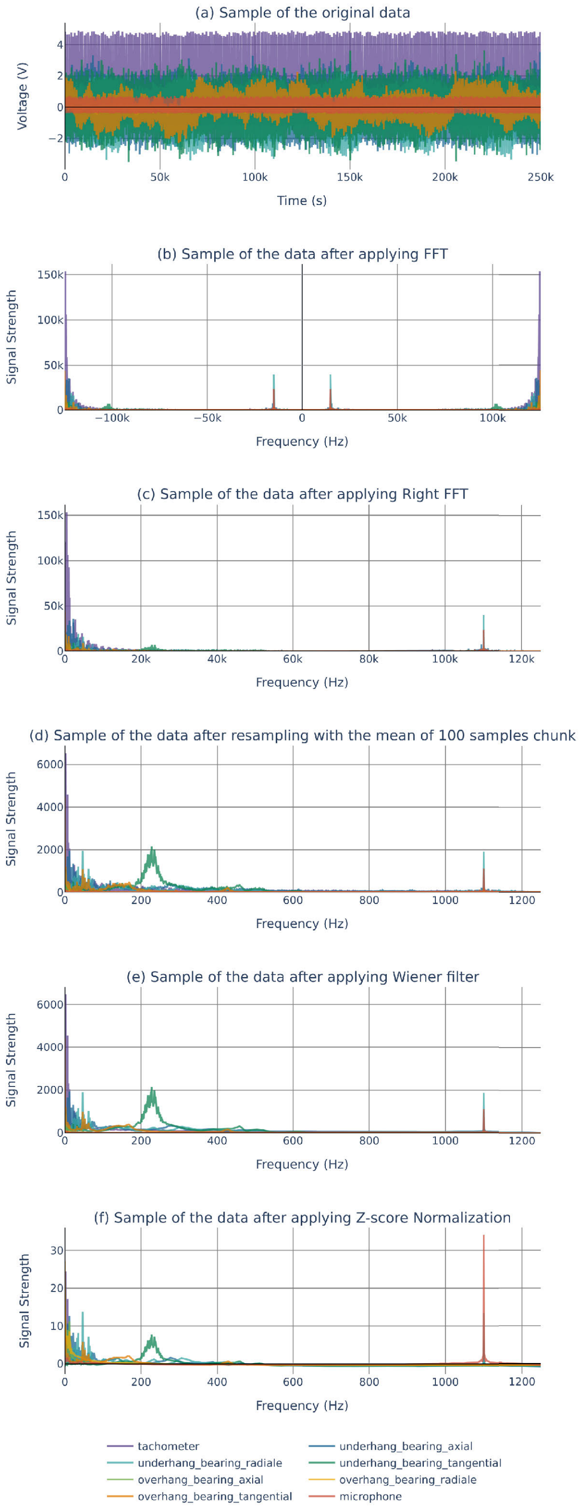

FIGURE 4. **Subsequent preprocessing phase over the data found inside (verticalmisalignment/1.90mm/51.2.csv file): (a) Sample of the original data, (b) Sample of the data after applying FFT, (c) Sample of the data after applying one side FFT (taking only the negative side by transforming its sign into positive), (d) Sample of the data after re-sampling with the mean of 100 samples chunk, (e) Sample of the data after applying Wiener filter, (f) Sample of the data after applying Z-score normalization.**

TABLE 6. Data column ranges before and after z-score normalization.

Column Name	Range Before	Range After
tachometer	15.726, 6476.568	-0.324, 24.467
underhang_bearing_axial	14.250, 1934.758	-0.791, 12.642
underhang_bearing_radiale	9.353, 1901.481	-0.681, 13.743
underhang_bearing_tangential	7.719, 2138.440	-0.518, 7.730
overhang_bearing_axial	0.407, 761.739	-0.207, 26.201
overhang_bearing_radiale	0.332, 165.388	-0.339, 20.419
overhang_bearing_tangential	2.337, 4457.891	-0.373, 27.296
microphone	5.926, 1116.069	-0.272, 34.079

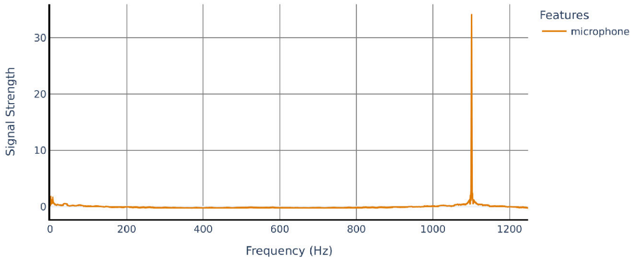

FIGURE 5. Sample of one microphone signal after applying all preprocessing steps.

mean of 2.8422e-17 and a standard deviation of 1. Using Equation 23, the obtained z-score for this frequency is 34.066, as clearly depicted in Figure 5, which specifically displays the microphone signal after the normalization step.

As a result of these subsequent prepossessing steps, the signals in Figure 4 (f) are obtained, which are the data intended to be fed to the proposed model.

VII. NUMERICAL AND MEASUREMENT RESULTS

In this section, we provide a comprehensive presentation of the experimental results, which encompass cross-validation architecture and its results, training and validation accuracies and losses, as well as a detailed analysis of the confusion matrix. Furthermore, we conduct an ablation study, involving six distinct operations, to explain the individual significance of each component within the proposed model. Previously, the main structure of the model is described in Figure 2, and its layers and their internal structures are mentioned in Table 1, and an overview of the proposed models' hyperparameters is presented in Table 2. The proposed model is firstly trained and validated using 90% of the data, reserving 10% for testing. As it can be noticed from Figure 6 the training and validation processes are implemented using a 5-fold cross-validation process.

As a result, at each iteration, the data are divided into 5 parts: 4 parts for training and one part for validation, each part containing 20% of the data. This arrangement results in 80% of the 90% of the data being used for training, and the remaining 20% for validation. Hence, during each iteration, the data for the training and validation processes keep changing, which allows for a clear assessment of the model's robustness five different times. The obtained validation accuracies are 100.0%, 99.7143%, 99.4269%, 99.4269%, and 99.4269%. respectively. Their losses are 0.8173, 0.8248, 0.8265, 0.8211, and 0.8277, respectively. As a consequence, the average validation accuracy is 99.599% with a standard deviation of +- 0.2293. As a result, the proposed models' overall train-validation-test accuracies are 99.87%, 99.599%, and 99.48%, respectively.

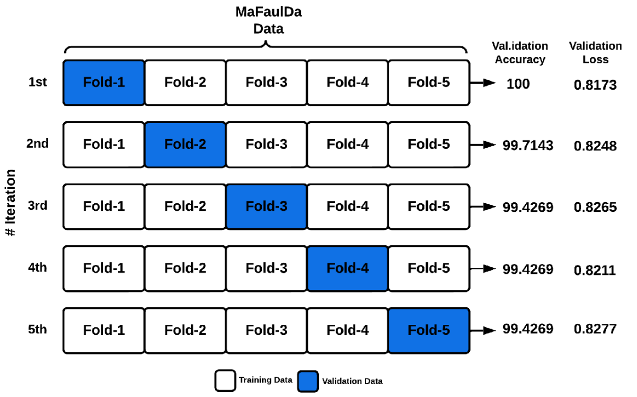

FIGURE 6. The architecture of a 5-Fold cross-validation approach.

In addition, the training and validation accuracy, along with the loss over epochs for each iteration, are visually represented in Figure 7, where part (a) corresponds to accuracy and part (b) to loss. The training processes are presented in blue, while the validation processes are indicated in red. It is apparent from Figure 7 that the proposed model exhibits remarkable consistency in its performance across different training and validation data, demonstrating its ability to achieve high accuracy and maintain low loss at each iteration. At the same time, it is evident that the model is trained over a smooth period and the model was learning, not memorizing, general features, as every increase in the training accuracy led to an increase in the validation accuracy and every decrease in the training loss is accompanied with a decrease in the validation loss. Moreover, it is obvious from the figure that the training loss was always under the validation one. This leads to the fact that the models' layers and their hyperparameters are precisely selected. As a result, it can be deduced that the proposed model is resilient to both underfitting and overfitting problems.

The confusion matrix of the obtained results is shown in Figure 8, which can be considered as a matrix that illustrates the number of correct and incorrect predictions for each class. Both the x-axis and y-axis of this matrix are the same and they show the name of the predicted classes. It is obvious that the model has a significant number of truly predicted values based on each class. It can completely predict 8 classes which are normal (no-fault), imbalance, overhang-ball, overhang-cage, overhang-outer race, underhang-ball, underhang-outer race, and underhang-cage. However, it only made one mistake as it predicted one sample of horizontal misalignment fault as a vertical misalignment fault.

Overall, the proposed model achieved an impressive accuracy of 0.9948, f1 score of 0.9949, precision of 0.9951, and recall of 0.9948 on the entire data. Table 7 provides a more detailed breakdown of the model's performance on

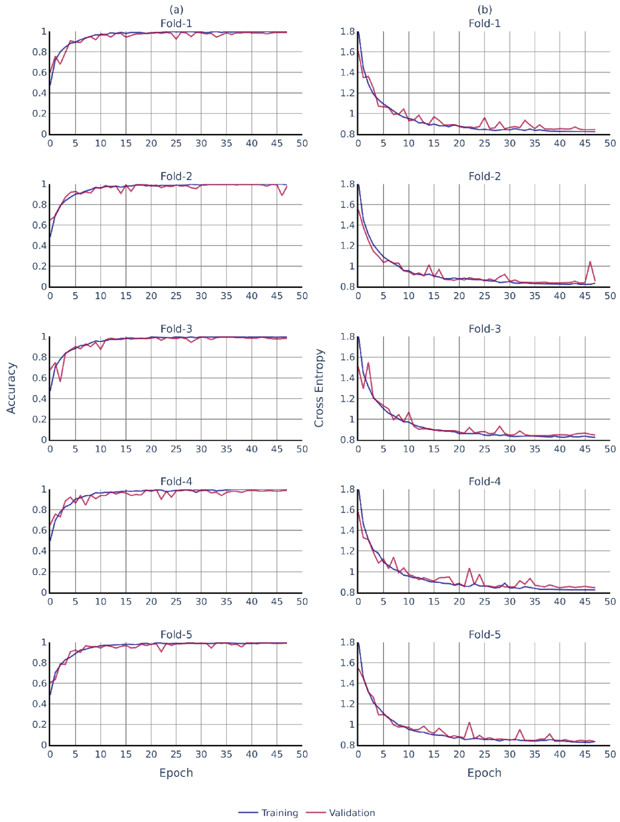

FIGURE 7. 5-Fold cross-validation results: (a) 5-fold cross-validation accuracy, (b) 5-fold cross-validation Loss.

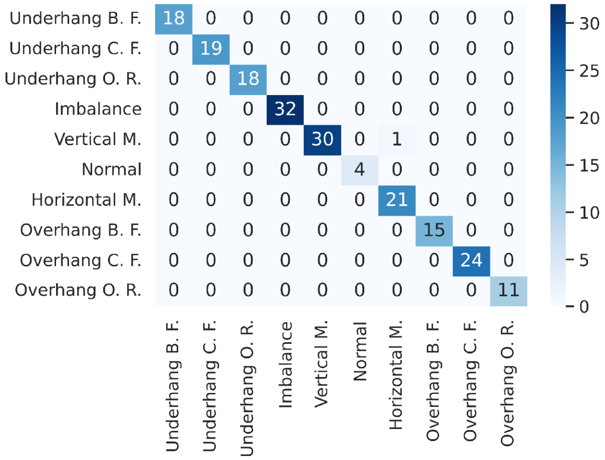

FIGURE 8. Output confusion matrix.

each class, including precision, recall, f1-score, and support (number of samples). This table shows that the model performs exceptionally well on all classes, with precision and recall scores consistently above 0.99. This suggests that the proposed model is able to accurately classify all of the different situations in the data.

Furthermore, the performance and importance of each component of the proposed model are solely investigated by conducting an ablation study. This involves removing some components while keeping others [39]. This is achieved by applying six different processes, as stated in Table 8, while the seventh process corresponds to the proposed model with all its components. It is clear that the FC2 layer is present in each process as it standardizes the parameters to match the number of predicted classes, 10 outputs. In the second and fourth situations, only LSTM and FC2 or LSTM, FC1, and FC2 are considered, and the obtained accuracy is the worst with equal accuracy, 95.36%. However, the trainable parameters in the fourth situation are bigger with no improvements in performance, making it the worst situation considered. When considering the first situation, only GRU and FC2 layers, the model achieved 97.42% accuracy. When the FC1 layer is added to the GRU and FC2 layers, the accuracy is enhanced to 98.97%, as it increases both the non-linearity and the feature combination operations. This can be considered as high accuracy, but it is lower than the achieved accuracy in different studies. On the other hand, adding an LSTM layer to the third situation leads to the fifth situation, where the accuracy is enhanced a little bit to 97.94% with almost double the number of complexity, which means double the training time. In the sixth situation, the GRU, FC1, LSTM, and FC2 layers are included, and the accuracy is only 0.9845, as the model's complexity is increased, and the model starts memorizing specific features. Based on that, L1 is added to the sixth situation, which is the proposed model. The model training process is enhanced as the model tries to focus on important features, moving away from specific features. The accuracies and losses of each situation are presented in Figure 9 (a) and (b), respectively. It is evident that the model, with its complete structure, exhibits the most relevant accuracy and loss charts, as they both increase and decrease smoothly. This suggests that the model consistently learns information with each epoch.

Consequently, the proposed model significantly outperforms all state-of-the-art models in terms of accuracy,

TABLE 7. Metric results for each class.

Classes	precision	recall	f1-score	support
Underhang Ball Fault	1.00	1.00	1.00	18
Underhang Cage Fault	1.00	1.00	1.00	19
Underhang Outer Race	1.00	1.00	1.00	18
Imbalance	1.00	1.00	1.00	32
Vertical Misalignment	1.00	0.97	0.98	31
Normal	1.00	1.00	1.00	4
Horizontal Misalignment	0.96	1.00	0.98	22
Overhang Ball Fault	1.00	1.00	1.00	15
Overhang Cage Fault	1.00	1.00	1.00	24
Overhang Outer Race	1.00	1.00	1.00	11

TABLE 8. Ablation study considered situations.

GRU	FC 1	LSTM	L1	FC 2	Parameters	Accuracy
✓				✓	801792	0.9742
		✓		✓	273930	0.9536
✓	✓			✓	868746	0.9897
	✓	✓		✓	305546	0.9536
✓		✓		✓	1591818	0.9794
✓	✓	✓		✓	1264266	0.9845
✓	✓	✓	✓	✓	1264266	0.9948

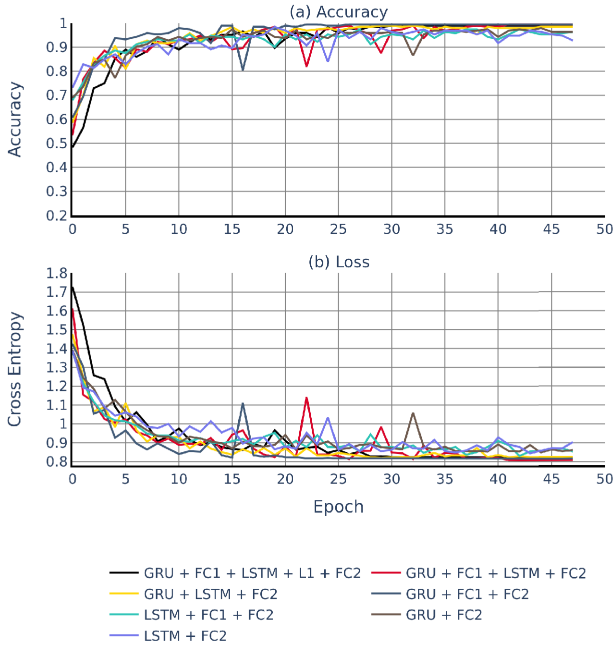

FIGURE 9. Ablation study results: (a) Considered model accuracy, (b) Considered model loss.

TABLE 9. Comparison with the state-of-the-art.

Study	Method	Classes	Accuracy
Marins, M. A. et al [6]	SBM	7	98.50
Souza, R. M. et al [16]	PdM-CNN	**10**	97.25
Ribeiro, F. et al [17]	SBM	5	96.40
Messaoudi, M. et al [18]	ANN-SMOTE	6	97.00
Shen, J. et al [19]	HNN	6	84.00
Alzghoul, A. et al [20]	ANN-SMOTE	6	97.10
Nath, A. G. et al [21]	LSTM, GRU	4	98.32
Nath, A. G. et al [22]	CNN-FRP, LSTM	4	99.00
Proposed Model	**GRU& FC&LSTM&FC**	**10**	**99.48**

as shown in Table 9. This table consists of four different columns which provide detailed information on the publication reference, the model used, the number of classified situations, and the obtained accuracies for each model, respectively. Apparently, the CNN-FRP, LSTM model in [22] achieves the closest accuracy to the proposed model, but it can only classify 4 different situations. In contrast, the proposed model can classify 10 different situations, which is the highest number of classifications reported in any publication, except [16]. This is a significant achievement, as it demonstrates that the proposed model is more accurate and versatile than any other model previously proposed for this task.

VIII. CONCLUSION

In summary, this research introduces a specially prepared RNN model for classifying 10 different motor faults using time-series data from the MaFaulDa database. The proposed model uses dual RNN layers which are GRU and LSTM with 512 and 256 neurons, respectively. These two layers are connected through an FC layer containing 128 neurons. The L1 regularization is applied over LSTM as well. Specific prepossessing steps such as using re-sampling technique, Wiener filter, and z-score normalization are applied on the original data. The model achieves impressive accuracy without data augmentation, outperforming other publications with train-validation-test accuracies of 99.87%, 99.599%, and 99.48%, respectively. This signifies the model's robustness and effectiveness in motor fault classification, with an overall accuracy of 99.48%, an F1 score of 0.9949, a precision of 0.9951, and a recall of 0.9948. In future work, the proposed models' complexity, and number of parameters, could be reduced. Moreover, the proposed model could be expanded to classify the subcategories, weights, and amounts of each fault as separate classes, thus increasing the number of identifiable classes from 10 to 42.

REFERENCES

[1] H. A. Toliyat and G. B. Kliman, *Handbook Electric Motors*, vol. 120. Boca Raton, FL, USA: CRC Press, 2018.

[2] A. Bingamil, I. Alsyouf, and A. Cheaitou, "Condition monitoring technologies, parameters and data processing techniques for fault detection of internal combustion engines: A literature review," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, Nov. 2017, pp. 1–5.

[3] S. Tang, S. Yuan, and Y. Zhu, "Deep learning-based intelligent fault diagnosis methods toward rotating machinery," *IEEE Access*, vol. 8, pp. 9335–9346, 2020.

[4] J. Liu, W. Wang, and F. Golnaraghi, "An enhanced diagnostic scheme for bearing condition monitoring," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 309–321, Feb. 2010.

[5] I. H. Ozcan, O. C. Devecioglu, T. Ince, L. Eren, and M. Askar, "Enhanced bearing fault detection using multichannel, multilevel 1D CNN classifier," *Electr. Eng.*, vol. 104, no. 2, pp. 435–447, Apr. 2022.

[6] M. A. Marins, F. M. L. Ribeiro, S. L. Netto, and E. A. B. da Silva, "Improved similarity-based modeling for the classification of rotating-machine failures," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1913–1930, Mar. 2018.

[7] J. B. Coble, "Merging data sources to predict remaining useful life—An automated method to identify prognostic parameters," Ph.D. dissertation, Dept. Eng., Univ. Tennessee, Knoxville, TN, USA, 2010.

[8] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 441–451, Mar. 2019.

[9] A. Gandhi, T. Corrigan, and L. Parsa, "Recent advances in modeling and online detection of stator interturn faults in electrical motors," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1564–1575, May 2011.

[10] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.

[11] R. B. Randall and J. Antoni, "Rolling element bearing diagnostics—A tutorial," *Mech. Syst. Signal Process.*, vol. 25, no. 2, pp. 485–520, 2011.

[12] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, Jul. 2016.

[13] (2016). *MaFaulDa—Machinery Fault Database*. Accessed: Sep. 6, 2023. [Online]. Available: https://www02.smt.ufrj.br/~offshore/mfs/page_01.html

[14] M. Maashi, N. Alwhibi, F. Alamr, R. Alzahrani, A. Alhamid, and N. Altawallah, "Industrial duct fan maintenance predictive approach based on random forest," in *Proc. 9th Int. Conf. Inf. Technol. Converg. Services (ITCSE)*, May 2020, pp. 177–184.

[15] H. Xue, H. Wang, P. Chen, K. Li, and L. Song, "Automatic diagnosis method for structural fault of rotating machinery based on distinctive frequency components and support vector machines under varied operating conditions," *Neurocomputing*, vol. 116, pp. 326–335, Sep. 2013.

[16] R. M. Souza, E. G. S. Nascimento, U. A. Miranda, W. J. D. Silva, and H. A. Lepikson, "Deep learning for diagnosis and classification of faults in industrial rotating machinery," *Comput. Ind. Eng.*, vol. 153, Mar. 2021, Art. no. 107060.

[17] R. M. Souza, E. G. Nascimento, U. A. Miranda, W. J. Silva, and H. A. Lepikson, "Deep learning for diagnosis and classification of faults in industrial rotating machinery," *Comput. Ind. Eng.*, vol. 153, pp. 2–10, 2021.

[18] M. Messaoudi, S. S. Refaat, M. Massaoudi, A. Ghrayeb, and H. Abu-Rub, "Classification of mechanical faults in rotating machines using SMOTE method and deep neural networks," in *Proc. 48th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2022, pp. 1–6.

[19] J. Shen, J. Chowdhury, S. Banerjee, and G. Terejanu, "Machine fault classification using Hamiltonian neural networks," 2023, *arXiv:2301.02243*.

[20] A. Alzghoul, A. Jarndal, I. Alsyouf, A. A. Bingamil, M. A. Ali, and S. AlBaiti, "On the usefulness of pre-processing methods in rotating machines faults classification using artificial neural network," *J. Appl. Comput. Mech.*, vol. 7, no. 1, pp. 254–261, 2021.

[21] A. G. Nath, A. Sharma, S. S. Udmale, and S. K. Singh, "An early classification approach for improving structural rotor fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[22] A. G. Nath, S. S. Udmale, D. Raghuwanshi, and S. K. Singh, "Improved structural rotor fault diagnosis using multi-sensor fuzzy recurrence plots and classifier fusion," *IEEE Sensors J.*, vol. 21, no. 19, pp. 21705–21717, Oct. 2021.

[23] E. G. S. Nascimento, J. S. Liang, I. S. Figueiredo, and L. L. N. Guarieiro, "T4PdM: A deep neural network based on the transformer architecture for fault diagnosis of rotating machinery," 2022, *arXiv:2204.03725*.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[25] A. A. E. Donkol, A. G. Hafez, A. I. Hussein, and M. M. Mabrook, "Optimization of intrusion detection using likely point PSO and enhanced LSTM-RNN hybrid technique in communication networks," *IEEE Access*, vol. 11, pp. 9469–9482, 2023.

[26] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600.

[27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[28] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," 2014, *arXiv:1409.1259*.

[29] Y. Alkhanafseh, M. Altin, A. Çakir, E. Karabiyik, E. Yildiz, and S. Akyüz, "Intelligent network monitoring system using an ISP central points of presence," in *Proc. Int. Conf. Intell. Fuzzy Syst.* Cham, Switzerland: Springer, Aug. 2023, pp. 246–254.

[30] S. Lenkala, R. Marry, S. R. Gopovaram, T. C. Akinci, and O. Topsakal, "Comparison of automated machine learning (AutoML) tools for epileptic seizure detection using electroencephalograms (EEG)," *Computers*, vol. 12, no. 10, p. 197, Sep. 2023, doi: 10.3390/computers12100197.

[31] D. M. W. Powers, "Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation," School Inform. Eng., Flinders Univ., Adelaide, SA, Australia, Tech. Rep. SIE-07-001, Dec. 2007.

[32] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.

[33] J. Antoni, "The spectral kurtosis: A useful tool for characterising non-stationary signals," *Mech. Syst. Signal Process.*, vol. 20, no. 2, pp. 282–307, Feb. 2006.

[34] J. Semmlow, "Stochastic, nonstationary, and nonlinear systems and signals," in *Circuits, Signals and Systems for Bioengineers* (Biomedical Engineering). New York, NY, USA: Academic Press, 2018, ch. 10, pp. 449–489.

[35] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction Wiener filter," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 14, no. 4, pp. 1218–1234, Jul. 2006.

[36] W. K. Pratt, "Generalized Wiener filtering computation techniques," *IEEE Trans. Comput.*, vol. C-21, no. 7, pp. 636–641, Jul. 1972.

[37] M. Z. Al-Faiz, A. A. Ibrahim, and S. M. Hadi, "The effect of Z-score standardization (normalization) on binary input due the speed of learning in back-propagation neural network," *Iraqi J. Inf. Commun. Technol.*, vol. 1, no. 3, pp. 42–48, Feb. 2019.

[38] P. S. Bimbhra and S. Kaur, *Power Electronics*, vol. 2. Delhi, India: Khanna, 2012.

[39] R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen, "Ablation studies in artificial neural networks," 2019, *arXiv:1901.08644*.

YOUSEF ALKHANAFSEH received the B.S. and M.S. degrees from the Electrical Engineering Department, Istanbul Technical University (ITU), in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree. He is also a Software Developer, particularly specializing in big data engineering with Turknet Communication Services Company, ?stanbul, Turkey. His research interests include creating big data clusters, processing large data, utilizing data streaming tools, deploying ML/NN models, designing workflow systems, managing cloud storages, working with databases, implementing containerization, and building pipelines.

TAHIR CETIN AKINCI (Senior Member, IEEE) received the bachelor's degree in electrical engineering, in 2000, and the master's and Ph.D. degrees, in 2005 and 2010, respectively. From 2003 to 2010, he was a Research Assistant with Marmara University, ?stanbul, Turkey. He was a Full Professor with the Electrical Engineering Department, Istanbul Technical University (ITU), in 2020. He was the Vice Dean of the Graduate School, from 2020 to 2021, and the Electrical and Electronics Engineering Faculty, from 2020 to 2021. He assumed the role of a Visiting Scholar with the University of California, Riverside (UCR). His research interests include artificial neural networks, deep learning, machine learning, cognitive systems, signal processing, power systems, and data analysis.

EMINE AYAZ received the B.S., M.S., and Ph.D. degrees from the Electrical Engineering Department, Istanbul Technical University (ITU), in 1993, 1997, and 2002 respectively. In 1999, she joined the Engineering Department and the Maintenance and Reliability Centre, University of Tennessee, to do research on accelerated aging studies of electric motors. She is currently a Full Professor with the Electrical Engineering Department, ITU. Her research interests include signal processing, soft computing, and condition monitoring in electric power systems.

ALFREDO A. MARTINEZ-MORALES received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Riverside (UCR), in 2005, 2008, and 2010, respectively. He is currently the Managing Director of the Southern California Research Initiative for Solar Energy (SC-RISE) and holds the position of a Research Professor with the Bourns College of Engineering, Center for Environmental Research and Technology (CE-CERT). He plays a key role as the Principal Investigator of the Sustainable Integrated Grid Initiative (SIGI) and the Distributed Energy Resources Laboratory (DERL), UCR, contributing to the engineering, permitting, and deployment of microgrids throughout Southern California. His current research interests include solar cells, alkali metal ion batteries, highly integrated renewables, energy storage systems, and microgrids.