mathematics

MDPI

Article

Electric Vehicle Motor Fault Detection with Improved Recurrent 1D Convolutional Neural Network

Prashant Kumar [1], Prince [2], Ashish Kumar Sinha [3] and Heung Soo Kim [1,*]

[1] Department of Mechanical, Robotics and Energy Engineering, Dongguk University, Seoul 04620, Republic of Korea; prashantkumar@dgu.ac.kr
[2] Department of Industrial & Systems Engineering, Dongguk University, Seoul 04620, Republic of Korea; prince91@dgu.ac.kr
[3] Department of Electrical Engineering, GLA University, Mathura 281406, India; ashish.sinha@gla.ac.in
* Correspondence: heungsoo@dgu.edu; Tel.: +82-2-2260-8577; Fax: +82-2-2263-9379

Abstract: The reliability of electric vehicles (EVs) is crucial for the performance and safety of modern transportation systems. Electric motors are the driving force in EVs, and their maintenance is critical for efficient EV performance. The conventional fault detection methods for motors often struggle with accurately capturing complex spatiotemporal vibration patterns. This paper proposes a recurrent convolutional neural network (RCNN) for effective defect detection in motors, taking advantage of the advances in deep learning techniques. The proposed approach applies long short-term memory (LSTM) layers to capture the temporal dynamics essential for fault detection and convolutional neural network layers to mine local features from the segmented vibration data. This hybrid method helps the model to learn complicated representations and correlations within the data, leading to improved fault detection. Model development and testing are conducted using a sizable dataset that includes various kinds of motor defects under differing operational scenarios. The results demonstrate that, in terms of fault detection accuracy, the proposed RCNN-based strategy performs better than the traditional fault detection techniques. The performance of the model is assessed under varying vibration data noise levels to further guarantee its effectiveness in practical applications.

Keywords: electric vehicle; recurrent convolutional neural network (RCNN); motor; vibration

MSC: 68T01

check for updates

Citation: Kumar, P.; P.; Sinha, A.K.; Kim, H.S. Electric Vehicle Motor Fault Detection with Improved Recurrent 1D Convolutional Neural Network. *Mathematics* **2024**, *12*, 3012. https://doi.org/10.3390/math12193012

Academic Editors: Adrian Olaru, Gabriel Frumusanu and Catalin Alexandru

Received: 2 September 2024
Revised: 19 September 2024
Accepted: 23 September 2024
Published: 26 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Electric vehicles (EVs) have long been at the forefront of modern transportation, owing to the ever-increasing requirement for sustainable and eco-friendly replacements for the traditional internal combustion engine vehicle in order to prevent increasing pollution levels and the resultant global warming. The EV is generally categorized into four types, namely the battery EV or pure EV (BEV or PEV), the plug-in hybrid EV (PHEV), the hybrid EV (HEV), and the fuel cell EV (FCEV) [1]. A typical EV drive train for all these classifications of EVs consists of numerous vital components that ensure its problem-free running. Figure 1 shows a schema of this [1]. Of all the components present in an EV, the most critical component is the electric motor. Recent studies have comprehensively shown that induction motors (IMs) are among the very frequently employed motors in the drive train of the conventional as well as hybrid EV [2–5]. Like any traditional IM used in various segments of contemporary industries, an IM employed in an EV is subjected to various undesirable stresses that can potentially hamper its general functioning, resulting in undue distress to the EV user. Avoiding this undesirable situation requires the employment of robust monitoring systems to ensure the reliability and efficacy of the induction motor, and, consequently, the overall drive train of the EV.

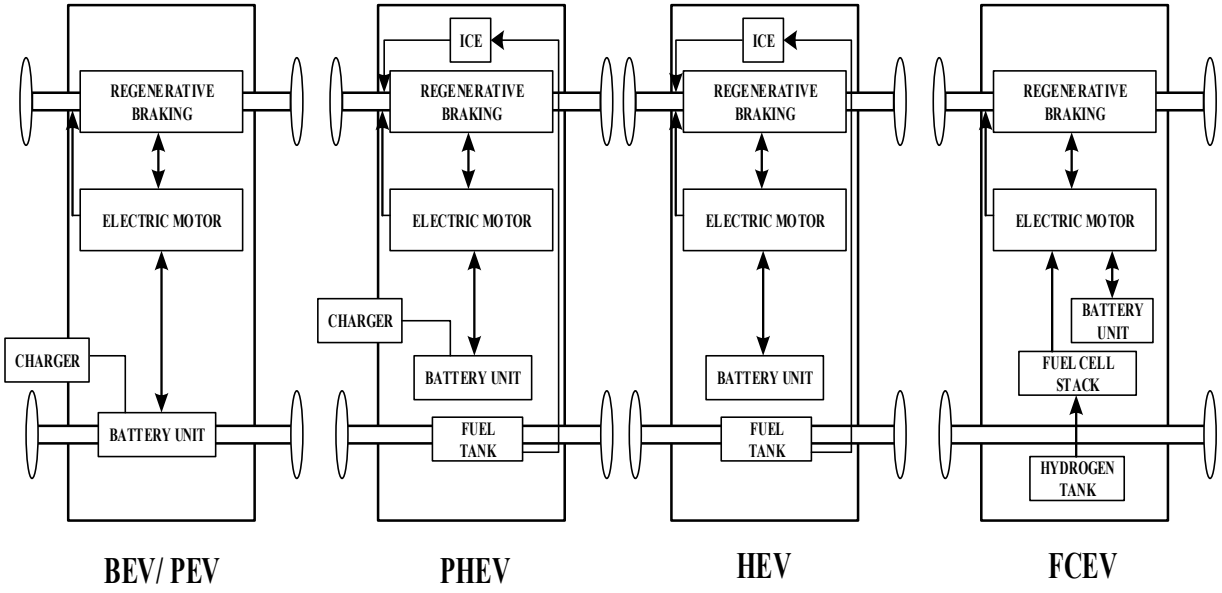

Figure 1. Different types of EVs.

Faults in IMs often trigger substantial performance deterioration, increased maintenance costs, and even detrimental failures of the EV. Consequently, the incipient and accurate detection and identification of these faults is essential to enhance the safety, longevity, and overall performance of the EV [6–8]. Recent studies have shown bearing damage to be one of the most recurrent and common faults to occur in the induction motor [9–11]. Figure 2 demonstrates the overall contribution of various faults that generally occur in an IM [12].

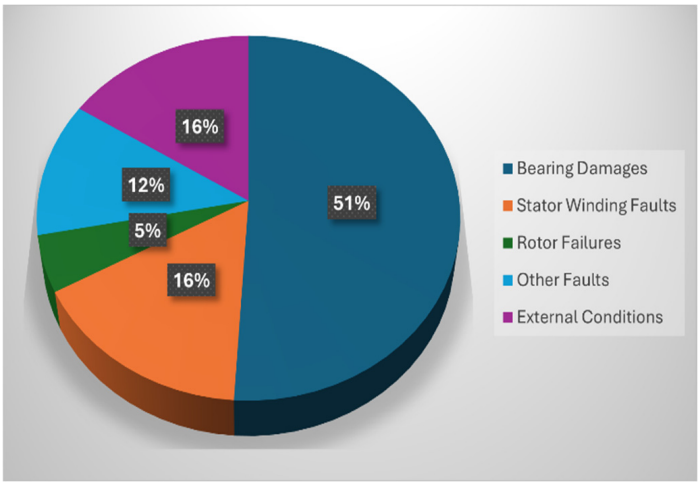

Figure 2. Contribution of different types of faults in induction motors [12].

Previous studies have extensively shown that bearing failure in the electric motor occurs mainly due to one of the following reasons: improper lubrication, inadequate or improper selection of the bearing, improper mounting of the bearing on the motor shaft, or manufacturing error. These prospective reasons result in various bearing faults and damage, which can be categorically classified as inner race defect, outer race defect, cage defect, and ball defect [13,14]. Figure 3 shows a schema of these regions of a typical ball-bearing that encounters the abovementioned defects:

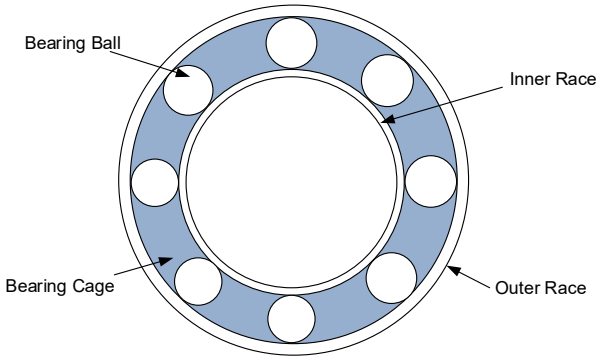

Figure 3. Schematic representation of a typical ball-bearing.

The existing traditional methods for fault detection in electric motors have often relied on model-based approaches or signal-processing techniques that can be complex and time-consuming [15,16]. This can be a major drawback owing to the application category of the IM, i.e., EV, which requires prompt detection or preferably incipient prediction of a prospective fault to avoid unwanted mishaps. With the advancement of artificial intelligence (AI) and deep learning, data-driven methods have emerged as robust tools for incipient fault detection. Among these, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have gained significant attention owing to their adeptness in modeling the spatial and temporal dependencies in the available dataset, respectively [17–19]. The DL-based fault detection approach has been significantly used in the health monitoring of industrial applications. Researchers [20] have used the CNN model for the bearing and rotor fault detection in the IMs. Researchers [21] have used the attention-based tree-inspired network for the health monitoring of the aero-engine bearings under noisy conditions. Researchers [22] have used the transfer-learning-based approach for servomotor bearing fault detection in industrial robots. Researchers [23] proposed a combination of a convolutional auto-encoder and convolutional block attention module for the fault diagnosis of the wind turbines. The developed method was used to diagnose wind turbine planetary gearboxes and bearings faults. Researchers [24] proposed an efficient strategy for the servomotor bearing fault diagnosis with the help of the DL approach based on the attention mechanism.

However, standard CNNs and RNNs have certain restraints in cases that involve applications to one-dimensional (1D) signals. These types of signals are dominant in the output of sensors that monitor the health of electric motors, for example, vibration signatures. To address these limitations, this paper intends to develop an improved 1D recurrent convolutional neural network (RCNN) architecture that is specifically tailored for fault detection in EV motors. The proposed model combines the strengths of both CNNs and RNNs, leveraging the convolutional layers to extract the local features from the 1D sensor signals and the recurrent layers, particularly the long short-term memory (LSTM) layers, to capture the temporal dependencies over time [25,26]. LSTM layers are well-adapted for this task because of their ability to effectively learn long-range dependencies in sequential data, which is crucial to detect subtle and complex patterns that are generally indicative of prospective bearing faults in the electric motor [27,28]. By integrating LSTM layers into the 1D-RCNN, the model can better handle time-series data, improving its accuracy in fault detection compared to the traditional methods [29] as well as the standalone CNN- or RNN-based fault detection methods. The main contributions of the proposed work are listed below:

- A combination of CNN and LSTM has been used to develop an efficient RCNN model for effective fault detection in EV motors.
- The proposed model has been evaluated on the experimental dataset to understand its efficacy.
- Performance evaluation under variable noise conditions has been implemented to demonstrate its applicability in real-world scenarios.

- Various performance parameters have been considered to show the performance of the model along with the confusion matrix and receiver operating characteristic (ROC) curve.
- A comparative study of the proposed model has been conducted with other baseline methods.

This paper presents a comprehensive analysis of the proposed 1D-RCNN model using vibration signals for testing and training purposes. The obtained results further demonstrate the efficacy of the developed model in detecting the bearing damages that are likely to occur in the electric motor of an EV. The practical applicability of the proposed model is further enhanced by conducting the model training and testing under varying vibrational data noise levels. The experimental results show that the improved 1D-RCNN outperforms the traditional fault detection (FD) methods, offering a promising solution to enhance the reliability and safety of EVs. An inclusive study of the present manuscript should help prospective readers to understand the process of bearing fault detection (BFD) in electric motors employed in EVs and to better comprehend the subject at hand.

This paper has been organized into six sections. Section 2 deals with the conceptualization of the RCNN. Section 3 covers the planned methodology. Furthermore, Section 4 demonstrates the experimental setup used in the proposed work, while Section 5 encompasses the results obtained from the experimental setup. Finally, Section 6 concludes the manuscript.

2. Recurrent Convolutional Neural Network (RCNN)

A 1D-RCNN model is a complex neural network architecture that combines the advantages of CNNs and RNNs to efficiently handle sequential data with temporal or spatial dependencies [30]. The 1D convolutional layers in this architecture function as the first step, where they utilize one-dimensional filters to extract local information, such as the patterns, trends, or periodicities present in the input sequence. Extracting local features is essential to collect intricate details inside certain sequence regions. After the convolutional layers, pooling layers, such as max pooling, are frequently used to decrease the dimensionality and emphasize the most important characteristics. This improves the computing efficiency while reducing the risk of overfitting. The convolved and pooled data are fed into the recurrent layers, commonly employing LSTM units or Gated Recurrent Units (GRUs). The recurrent layers excel at representing long-term connections and temporal changes by continuously updating a hidden state while incrementally processing the sequence [31]. This allows them to capture the contextual linkages and sequential dependencies of the data [32]. After iterative processing, fully connected (dense) layers are commonly appended to consolidate the acquired information and execute the ultimate prediction or classification tasks. This is accomplished by employing activation functions, such as SoftMax for classification, or linear activations for regression tasks. This combination enables the network to extract local patterns using convolutional operations and subsequently comprehend the wider temporal or sequential context using recurrent processing. The 1D-RCNN model has various applications, including time series forecasting, speech recognition, natural language processing tasks, and health monitoring [33–36].

The model can predict future values in time series forecasting by comprehending short-term patterns and long-term trends. Speech recognition processes raw audio signals by extracting acoustic features and modeling temporal dependencies. Natural language processing tasks like sentiment analysis or sequence tagging benefit from capturing local word features and contextual relationships. Additionally, 1D-RCNNs can be used in health monitoring applications, such as analyzing ECG signals to detect anomalies or predict health conditions [37]. The 1D-RCNN offers the benefit of effectively capturing both the local and global dependencies in sequential data, making it highly effective for jobs requiring the comprehension of detailed patterns and overall sequences [38]. Nevertheless, this architectural design also poses obstacles, such as heightened computational intricacy and training complexity resulting from the amalgamation of the convolutional

and recurrent layers. Moreover, smaller datasets pose an increased risk of overfitting, requiring meticulous regularization and model adjustment. The 1D-RCNN model provides a strong and adaptable framework for addressing various sequential data processing tasks by utilizing the combined advantages of CNNs and RNNs, notwithstanding the difficulties involved [39,40].

The effectiveness of a 1D-RCNN is particularly notable in FD, particularly in industrial and mechanical systems where the analysis of time-series data is essential to identify anomalies or failures [41]. The 1D-RCNN utilizes a hybrid design to effectively capture local and long-term dependencies in sequential data. This includes vibration signals, temperature measurements, or other sensor data streams used to monitor the health of equipment. The 1D convolutional layers in the network serve as the initial processing stage, where they utilize filters to scan the input sequences and identify local patterns, such as abrupt spikes, irregular fluctuations, or repetitive noise, which may suggest early indications of a defect. These layers are highly proficient at detecting minor localized irregularities within the data that could be overlooked by less sophisticated detection methods. Once the local features have been extracted, the data are processed through pooling layers to compress the most important information. This process helps to eliminate unwanted noise and concentrate on the most significant components of the signal. The output generated by these layers is further used as an input regarding the recurrent layers, such as LSTM or GRUs, specifically designed to capture the data's temporal dependencies and sequential characteristics. The recurrent layers play a vital role in identifying faults by preserving and updating a recollection of past states while processing the sequence. This enables the network to detect patterns that evolve over time, such as a gradual decline in performance, or a breakdown. The 1D-RCNN is well-suited for detecting problems that develop gradually, or have long-term predecessors, due to its ability to track and establish connections between events occurring at diverse intervals. In the last phase of the network, fully connected layers combine the acquired data from the convolutional and recurrent layers. These combined features are then used to determine the existence or probability of a fault, usually through a classification or regression method. In FD, this choice may be expressed as an alarm, a probability score that quantifies the possibility of a fault occurring, or a classification label that identifies the specific type of fault observed. Using the 1D-RCNNs in defect detection has notable benefits, such as the capacity to autonomously acquire intricate characteristics from unprocessed sensor data without requiring manual feature extraction, which is frequently a laborious and error-prone procedure. Moreover, the ability of the network to manage the local irregularities and long-term patterns within the same structure makes it a potent tool for real-time monitoring systems. This is crucial as it enables the early identification of defects, preventing expensive downtime and prolonging the lifespan of the equipment. Nevertheless, implementing 1D-RCNNs for FD comes with certain difficulties, including the requirement for significant computational resources during training and the potential problem of overfitting, especially when the data are scarce or unevenly distributed across various fault categories. Although faced with several obstacles, the capacity of the 1D-RCNN to combine the advantages of convolutional and recurrent neural networks renders it a sophisticated and efficient method for FD. It offers strong and dependable performance in the supervision and upkeep of intricate industrial systems.

3. The Proposed Methodology

The application of DL has paved the way for an efficient BFD of the EV's electric motor with minimal requirements or no requirement of feature engineering. The proposed approach follows the hybrid approach, in which a combination of CNN and recurrent network, specifically the LSTM, has been developed for an efficient BFD. This approach utilizes the strengths of CNN in extracting the local features and LSTMs in acquiring the temporal dependencies. The first step in the model development involves data preparation. Let $\boldsymbol{x}(\boldsymbol{t})$ denote the time series vibration signal with N time steps and F features. The input

data are segmented into a length T, defined by the window length, extracted with a stride for overlapping windows.

$$\mathbf{X}=\begin{pmatrix} \mathbf{x_{1,1}} & \mathbf{x_{1,2}} & \cdots & \mathbf{x_{1,F}} \\ \mathbf{x_{2,1}} & \mathbf{x_{2,2}} & \cdots & \mathbf{x_{2,F}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_{T,1}} & \mathbf{x_{T,2}} & \cdots & \mathbf{x_{T,F}} \end{pmatrix} \tag{1}$$

The categorical fault labels $\boldsymbol{y}\boldsymbol{\epsilon}\{0,1,\ldots,\boldsymbol{C}-1\}$ are encoded with the help of one-hot encoding. For $\boldsymbol{C}$ representing the number of classes, the one-hot-encoded label vector $\boldsymbol{y_{label}}$ for class $\boldsymbol{c}$ can be expressed as

$$\mathbf{y_{label}}=[\mathbf{0},\ldots\mathbf{0},\mathbf{1},\mathbf{0},\ldots\mathbf{0}]^{\mathbf{T}} \quad (\mathbf{1\ at\ position\ c}) \tag{2}$$

The proposed RCNN model architecture combines the convolutional layers for extracting features and LSTM layers for temporal pattern identification. CNN can autonomously extract the hierarchical features from the input vibration data. It captures the local dependencies and patterns, such as trends or periodicity, by applying the filters across the vibration data. For a 1D convolution, the input is a time-series segment $\boldsymbol{X}$ of shape $\boldsymbol{T}\times\boldsymbol{F}$, where $\boldsymbol{T}$ signifies the quantity of time steps and $\boldsymbol{F}$ indicates the quantity of features. The convolution operation applies a filter $\boldsymbol{W}$ of size $\boldsymbol{k}$ over the input. The output of the convolutional operation at each position is expressed as

$$\mathbf{z(t)}=(\mathbf{W}*\mathbf{X})\mathbf{(t)}=\sum_{\mathbf{i=0}}^{\mathbf{k-1}}\mathbf{W(i).X(t+i)+b} \tag{3}$$

where b signifies the bias term, and $*$ denotes the convolution operation. A feature map is produced by repeating this process while the filter runs through the complete input sequence.

After the convolution operation, a rectified linear unit (ReLU) is commonly used as an activation function and introduces the non-linearity into the model, assisting it to learn more complicated patterns [18]. This is expressed as

$$\mathbf{a(t)=ReLU(z(t))=max(0,z(t))} \tag{4}$$

Batch normalization is employed to improve the training speed and stability; it sets the mean and variance of each layer's output to zero [42]. It has the following mathematical expression:

$$\mathbf{\hat{a}(t)}=\frac{\mathbf{a(t)}-\mu_{\mathbf{batch}}}{\sqrt{\sigma^{\mathbf{2}}_{\mathbf{batch}}+\boldsymbol{\varepsilon}}} \tag{5}$$

where $\boldsymbol{\mu_{batch}}$ denotes the mean, $\boldsymbol{\sigma^2_{batch}}$ denotes batch variance, and $\boldsymbol{\varepsilon}$ represents a small constant that prevents division by zero.

An LSTM network is a type of RNN designed to find long-term dependencies in sequence data. These networks are useful for learning temporal patterns in time-series data because they solve the vanishing gradient issue of conventional RNNs [31]. The LSTM cell contains multiple gates that control the flow of information. These comprise the forget gate, input gate, cell state update, and output gate.

$$\mathbf{forget(f_t)=\sigma(W_f\cdot[h_{t-1},x_t)+b_f)} \tag{6}$$

$$\mathbf{i_t=\sigma(W_i.[h_{t-1},x_t)+b_i)} \tag{7}$$

$$\mathbf{candidate\ vector}\left(\tilde{\mathbf{C}}_\mathbf{t}\right)\mathbf{=tanh(W_C\cdot[h_{t-1},x_t)+b_C)} \tag{8}$$

$$\mathbf{cell\ state\ update(C_t)=f_t\odot C_{t-1}+i_t\odot\tilde{C}_t} \tag{9}$$

The output of the LSTM cell is provided by

$$\mathbf{forget}(\mathbf{o_t})=\sigma(\mathbf{W_o}\cdot[\mathbf{h_{t-1}},\mathbf{x_t})+\mathbf{b_o}) \quad (10)$$

The LSTM can identify long-term dependencies in the time-series data by selectively remembering or forgetting information thanks to this series of procedures. CNNs and LSTMs are combined, affording the model the advantages of both architectures:

- From the input time series, CNNs effectively extract local features like edges, spikes, or patterns inside very brief time frames.
- Considering that the data are sequential, LSTMs can capture the temporal dependencies between these derived features.

After the LSTM layers, the output is flattened and passed through fully connected (dense) layers. A dense layer applies a linear transformation, followed by the non-linear function, and is mathematically expressed as

$$\mathbf{y_{dense}}=\mathbf{ReLU}(\mathbf{W_d}\mathbf{H_{flat}}+\mathbf{b_d}) \quad (11)$$

The network is able to study high-level input interpretations thanks to this layer, which combines the elements that the CNN and LSTM layers learned. To avoid overfitting, dropout is used during training to arbitrarily set some of the input units to zero. The dropout rate determines the percentage of units to drop. A dense layer with a SoftMax activation function comprises the last layer:

$$\mathbf{y_{out}}=\mathbf{Softmax}(\mathbf{W_o}\mathbf{y_{dense}}+\mathbf{b_o}) \quad (12)$$

The result is transformed into a probability distribution over the classes via the SoftMax function, where each value denotes the likelihood that the input belongs to a specific class. Figure 4 displays the architecture, and Table 1 shows the structure of the developed RCNN model:

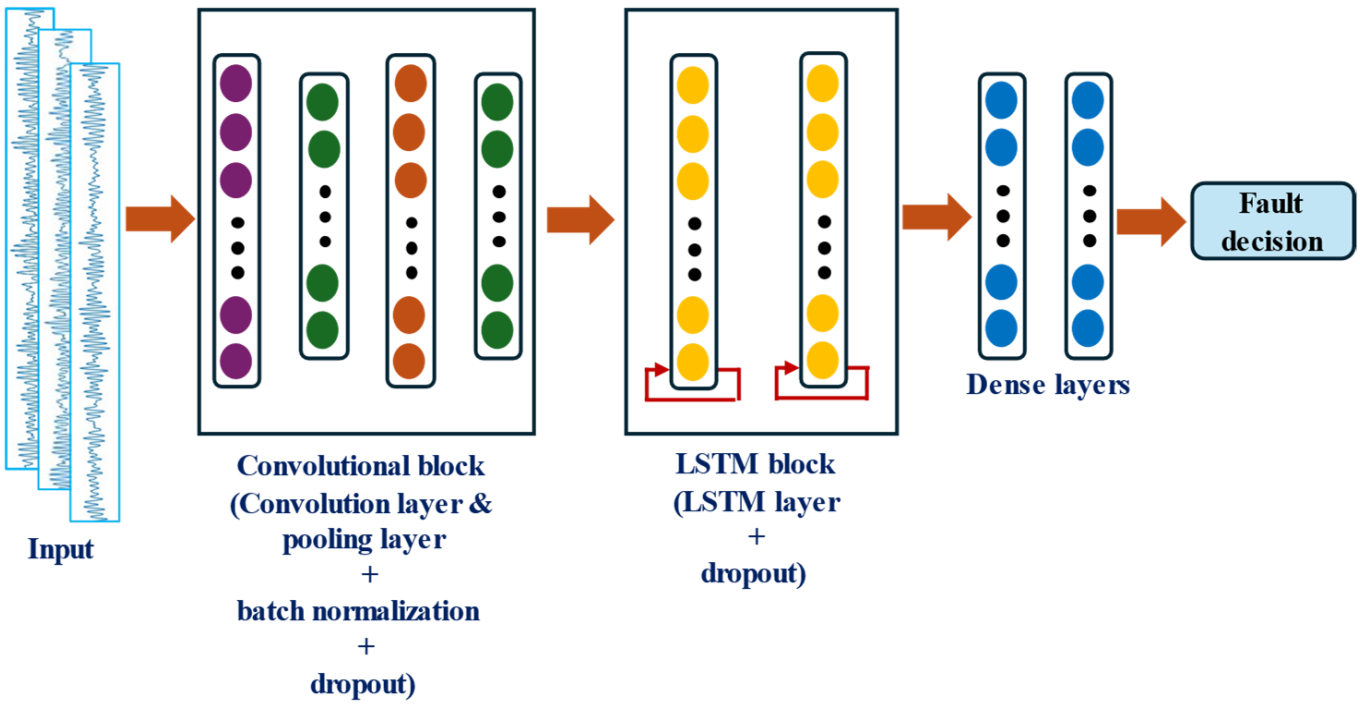

Figure 4. Schematic of the proposed RCNN model.

Table 1. Structure of the RCNN model.

Layer	Type	Filters	Kernel Size	Stride	Padding	Activation	Dropout
Input Layer	Input	-	-	-	-	-	-
Conv1D Layer 1	Convolutional	64	3	1	Valid	ReLU	-
Batch Normalization Layer 1	Normalization	-	-	-	-	-	-
Conv1D Layer 2	Convolutional	128	3	1	Valid	ReLU	-
Batch Normalization Layer 2	Normalization	-	-	-	-	-	-
LSTM Layer 1	Recurrent	128	-	-	-	-	-
Dropout Layer 1	Dropout	-	-	-	-	-	0.3
LSTM Layer 2	Recurrent	64	-	-	-	-	-
Dropout Layer 2	Dropout	-	-	-	-	-	0.3
Flatten Layer	Flatten	-	-	-	-	-	-
Dense Layer 1	Fully Connected	128	-	-	-	ReLU	-
Dropout Layer 3	Dropout	-	-	-	-	-	0.5
Output Layer	Fully Connected	num_ classes	-	-	-	SoftMax	-

4. Experimental Setup

The experimental setup comprises a 2 hp electric motor, torque transducer/encoder, dynamometer, and control electronics [43]. Figure 5 shows a schematic of the layout of the experimental setup [44]. To simulate different bearing failures, single-point flaws were introduced into the test bearing by employing electro-discharge machining. The test bearings support the motor shaft. An accelerometer was employed to gather vibration data at 48 kHz sampling frequency under various operating conditions. The vibration data were obtained for different bearing fault conditions, including bearing inner race fault (FIR), outer race fault (FOR), and faults in bearing balls (FBB). The vibration data were gathered under variable operating conditions and varying speeds at the motor drive end. Based on this, a dataset comprising four labels (FIR, FOR, FBB, and Normal (N)) and corresponding vibration data was developed. This dataset was then used to train, validate, and test the proposed RCNN model. The complete analysis was carried out on a PC with an i-7 Intel processor, NVIDIA GeForce RTX3080 Ti graphics card, and 32 GB memory.

The RCNN model was developed on these vibration data after data conditioning. Data conditioning is performed to ensure that the data are clean, and that there are no missing values. It helps to ensure efficient model training. Before feeding to the proposed RCNN model, the vibration data were segmented into multiple segments of fixed length. Finally, the prepared data were segregated into the training, validation, and test datasets. Multiple phases are involved in processing vibration segments with the proposed RCNN model, which combines the advantages of RNNs and CNNs. This method is deployed as it is efficient in examining time-series data like vibration signals since it can capture the temporal and local features needed for fault detection. The first stage of the RCNN model involves the convolution layers. This layer uses convolutional filters across the input data to extract the local patterns and features. Following the convolution process, non-linearity is introduced by applying an activation function (ReLU), which improves the model's capacity to identify intricate data patterns. This is then followed by max-pooling layers to reduce the dimensionality and highlight the most vital features. The output is sent to the recurrent layers, which include the LSTM layers after the convolutional layers have finished extracting the local features. The temporal connections between the features, which are essential to comprehend the sequential nature of vibration signals, are captured by these layers. The model can identify patterns that emerge over time because the recurrent layers

process the feature sequence and preserve a recollection of prior states. This is especially helpful in vibration analysis since flaws can appear as distinct vibrational sequences rather than as separate events. These features are processed further by dense layers. The Adam optimizer and the categorical cross-entropy loss function were employed during the RCNN model training. Adam was selected because it can adjust the learning rate during training and work effectively in noisy conditions. This is a multi-class classification problem, so categorical cross-entropy is applied. During training, performance is monitored using accuracy as the assessment metric. The training parameters of the proposed model are provided in Table 2. The proposed RCNN model efficiently manages vibration segments by utilizing convolutional layers to mine local information and recurrent layers to acquire temporal correlations. This strategy provides the model the ability to precisely identify and categorize faults, which makes it an effective tool for fault detection. Figure 6 shows a schematic of the layout of the fault detection and maintenance decision.

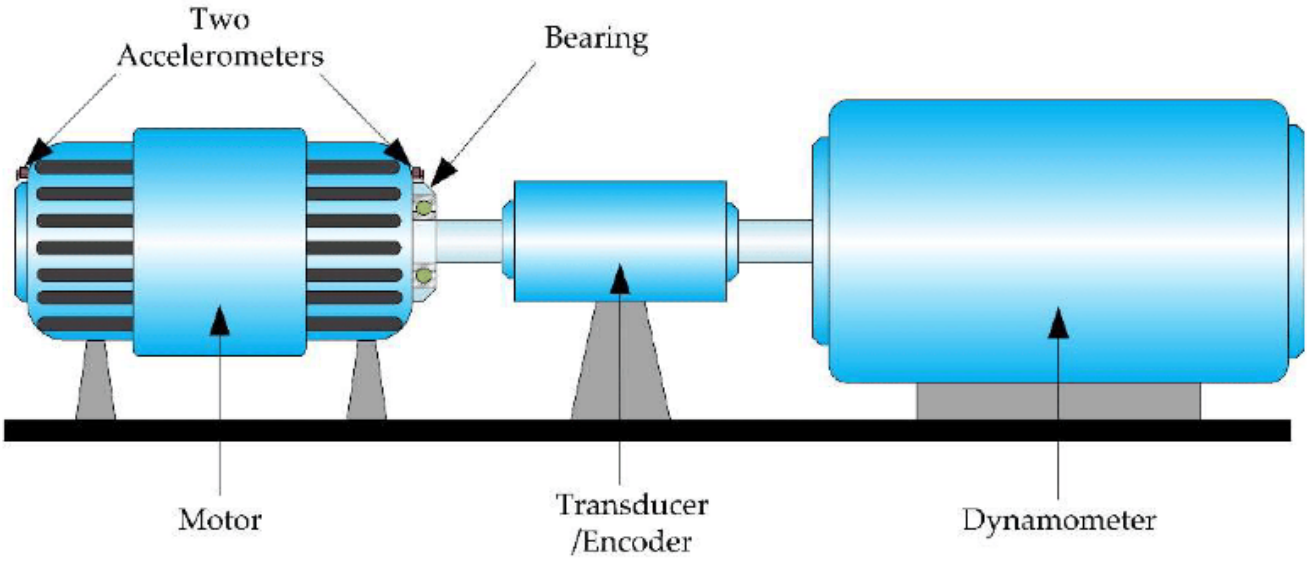

Figure 5. Schematic layout of the experimental setup.

Table 2. Training parameters of the proposed model.

Parameters	Values
Number of epochs	100
Batch size	50
Learning rate	0.01
Optimizer	Adam
Loss function	Categorical cross-entropy
Metrics	Accuracy
Overfitting handling	Dropout (30% in LSTM layers, 50% in Dense), Early Stopping (patience = 50), Batch Normalization

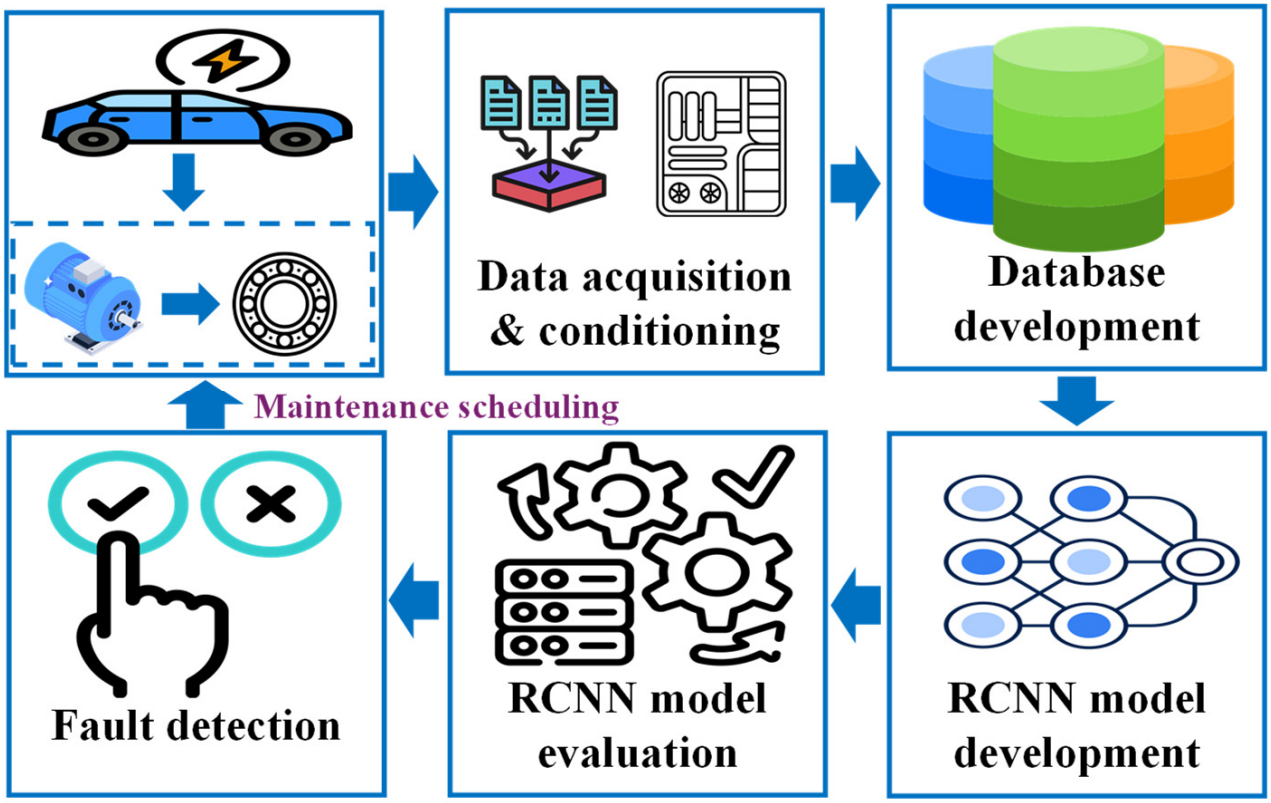

Figure 6. Schematic layout of the EV motor fault detection and maintenance decision.

5. Results and Discussion

5.1. Results Analysis

The efficient functioning of the EV's electric motor is crucial for the desired performance of the EV. It is prone to bearing faults, which are among the most common faults in motors, as previously discussed. The proposed RCNN model utilizes an innovative approach of amalgamating the convolutional layers with the LSTM layers for efficient BFD in the electric motor. The RCNN model was rigorously trained, validated, and tested on the electric motor bearing dataset to ensure optimal performance. The model hyperparameters were optimized using the random search method. Compared to a grid search, this method frequently identifies a good combination more quickly, especially when only a portion of the hyperparameters significantly affect the model's performance. During training, the performance is monitored using accuracy as the assessment metric. Each epoch involves training the model on the training set and validating its performance. The model's performance on the training set of data is used to calculate the training loss and accuracy. The validation loss and accuracy are also calculated. The process repeats for 100 epochs or until early stopping is triggered. Several dropout layers are implemented on distinct parts of the model to mitigate the likelihood of overfitting. For example, 50% dropout is used after the first dense layer, and 30% dropout is applied after each LSTM layer. Early stopping is used to halt training when the validation loss stops improving for 50 epochs. Pausing the model before it begins to memorize the training data helps to prevent the model from overfitting. After the convolutional layers, batch normalization is applied to normalize the activations. This enhances generalization and helps to stabilize and accelerate the training process. Figures 7 and 8 show the training and validation loss and accuracy curves, respectively. It is apparent from the curves that the RCNN model has converged in 85 epochs. It is evident from these curves that the model is starting to settle after 60 epochs. At around 85 epochs, the training and validation losses are essentially constant at the same value, which demonstrates that the model has high generalization capabilities.

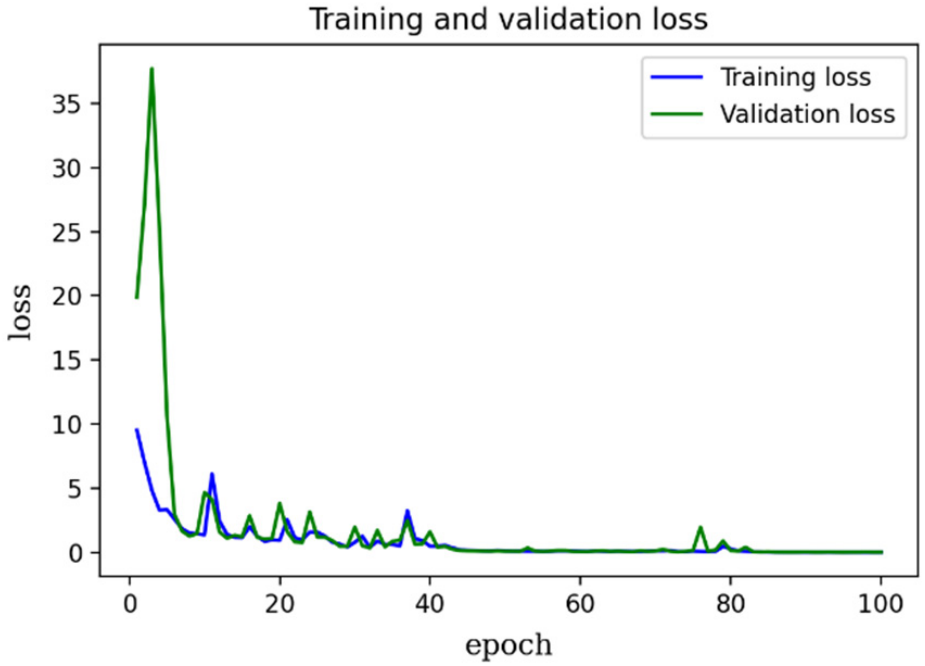

Figure 7. Loss curves of the proposed RCNN model on *the* training and validation datasets.

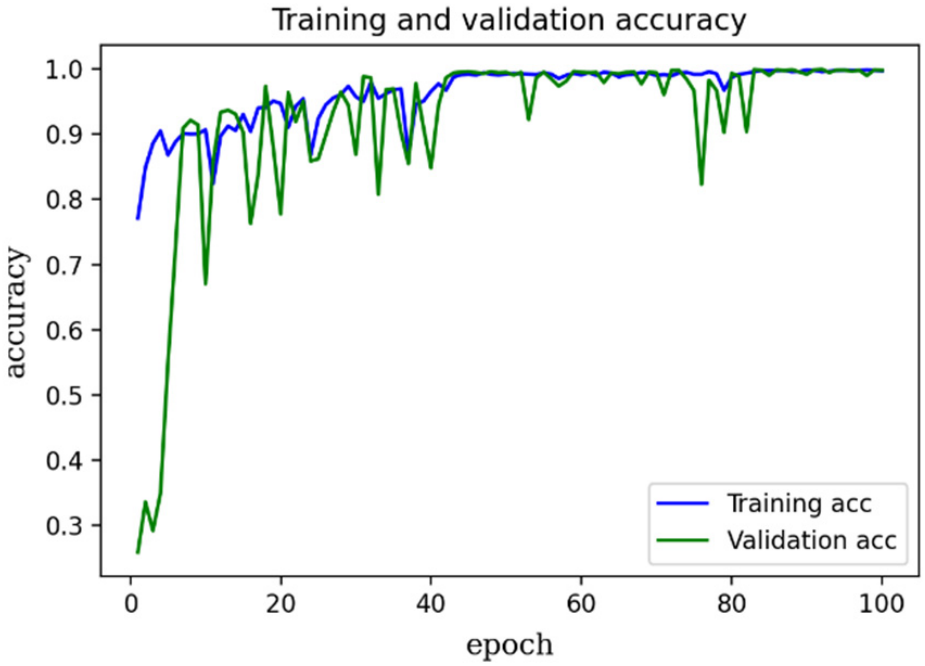

Figure 8. Accuracy curves of the proposed RCNN model on the training and validation datasets.

Once the model is trained, it can be used on unseen data to make predictions. The input to the model remains as the time-series data. The convolution, LSTM, and dense layers are traversed by the newly entered data. After processing the input, each layer passes the features it has extracted to the following layer. The last layer returns the probability for each class using the SoftMax activation. The outcome is a probability distribution over the potential classes (the total number of classes). The class with the highest probability is the one that is predicted. The performance of the proposed RCNN model is also assessed using performance parameters like precision (**p**), recall (**r**), and F1-score. The precision of a prediction is the percentage of actual positive forecasts among all the instances that the

model predicts as positive. Recall, sometimes referred to as sensitivity, gauges the accuracy of the model in identifying every pertinent instance (true positives) in a class. It symbolizes the proportion of real positive cases that the model correctly identified. The F1-score is a single statistic that provides a balance between precision and recall, calculated as the harmonic mean of the two. Also, the accuracy of the RCNN model is considered to assess the performance. Mathematically, these parameters can be expressed as

$$\mathbf{acc}=\frac{\mathbf{tp+tn}}{\mathbf{tp+tn+fp+fn}} \tag{13}$$

$$\mathbf{p}=\frac{\mathbf{tp}}{\mathbf{tp+fp}} \tag{14}$$

$$\mathbf{r}=\frac{\mathbf{tp}}{\mathbf{tp+fn}} \tag{15}$$

$$\mathbf{F1}=\frac{\mathbf{2\times p\times s}}{\mathbf{p+s}} \tag{16}$$

where variables tp, tn, fp, and fn represent actual positive trials, actual negative trials, and incorrect positive and negative trials, respectively.

The four different states of the motor bearing have been considered, namely FIR, FOR, FBB, and normal (N) state. Table 3 shows the different performance parameters. It is evident from Table 2 that the performance parameter values are on the higher side, indicating the superior performance of the model. Also, Figure 9 shows the confusion matrix (CFM) to reveal the RCNN model's performance. When assessing a classification model's performance, particularly when dealing with multi-class classification, the CFM is an important notion to understand. When a model is applied to a set of test data whose true values are known, this table is frequently used to demonstrate how well it works. It is also evident from the CFM (Figure 9) that all the bearing states, namely FIR, N, FOR, and FBB, have been efficiently classified. Also, the receiver operating characteristic (ROC) curve is presented to demonstrate the performance of the model. The ROC curve offers a comprehensive visual representation of the performance of a model over all the potential classification criteria. It enables a thorough analysis of the trade-offs between these two metrics by plotting the true positive rate (sensitivity) versus the false positive rate (specificity). The total capacity of the model to distinguish the positive and negative classes is summarized by a single scalar value called the area under the curve (AUC). A higher AUC implies superior model performance. Figure 10 shows the ROC curve, and also the AUC values. In addition, the AUC values are higher, which supports the efficacy of the RCNN model.

Table 3. Performance parameters of the proposed RCNN model.

State	*p*	*r*	*F*1-Score
FBB	0.99	0.99	0.99
FIR	1.0	1.0	1.0
FOR	0.99	0.99	0.99
N	1.0	1.0	1.0

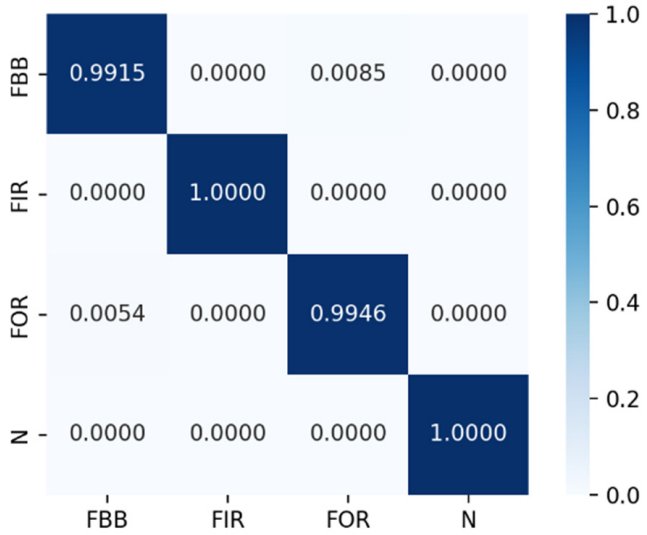

Figure 9. CFM for the proposed RCNN model.

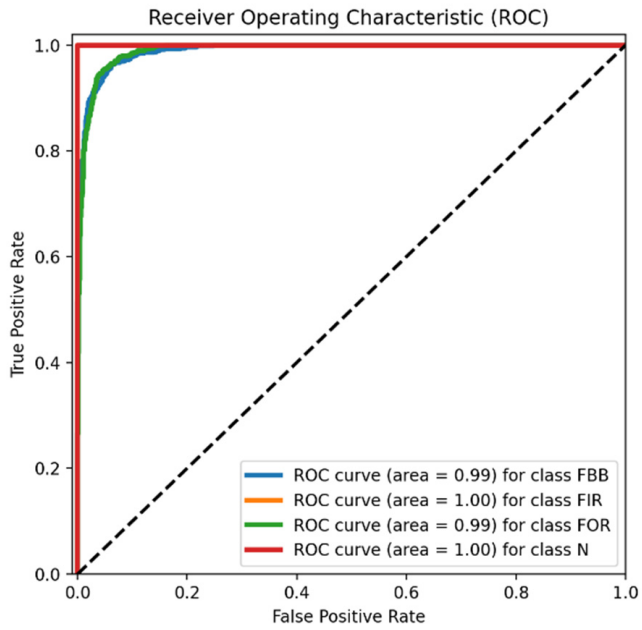

Figure 10. ROC curve for the proposed RCNN model.

5.2. Performance Evaluation under Noisy Conditions

In real-world applications, evaluating the performance of a proposed model under noisy conditions, especially in the context of EV motor BFD, is vital to determining the robustness and reliability of the model. This can be achieved through varying the signal-to-noise ratio (SNR), which emulates the real-world scenario where data may be contaminated with noise. This helps in understanding the model's performance despite the noisy data. In real-world applications, sensor data obtained from the EV motor bearings are often subjected to various sources of noise. A metric known as the SNR compares the strength of the required signal to the strength of ambient noise. This is generally expressed in decibels (dB) and mathematically expressed as

$$\mathbf{SNR(dB)=10\times log_{10}\frac{Signal\ power}{Noise\ power}} \tag{17}$$

White Gaussian noise has been added to the vibration signals for the analysis to simulate the noisy environment. In the fault detection domain, SNR values varying in the

range (−6 to 9) dB are generally considered for the analysis. Repeated experiments were carried out, with the structures and parameters of the proposed RCNN model remaining unchanged. Figure 11 displays the model's performance at various SNR levels. According to the study, the classification accuracy of the proposed model usually improves as the SNR levels rise. A higher SNR implies lower noise levels. The RCNN model performs almost as well as its baseline accuracy at higher SNR values, with little to no performance deterioration. If the SNR values are low, the performance has decreased slightly. It is evident from Figure 11 that, when the SNR value is −6 dB, the accuracy is 85.54%, and, when the SNR value is gradually increased to 9 dB, the accuracy of the proposed model becomes 98.01%. It is evident from Figure 11 that, when the SNR value rises, the performance of the model improves. This is obvious because, as the SNR value rises, the noise levels are less, and the performance is better. If the SNR value is less, the performance deteriorates due to the high level of noise in the data. However, it is still reasonable for an EV motor FD. The noise robustness of the proposed RCNN model has been comprehended by methodically adjusting the SNR and assessing the model's performance. This study helps to recognize that the RCNN model is highly effective in clean or noisy environments.

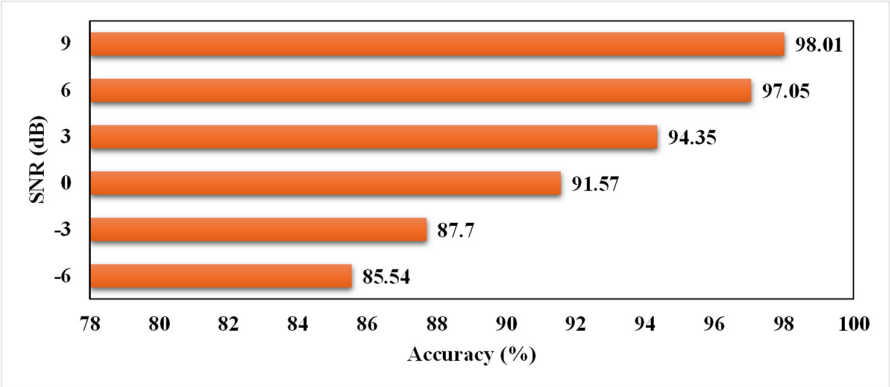

Figure 11. Performance of the proposed RCNN model under different SNR values.

5.3. Comparison with the Baseline Methods

The purported performance of the RCNN model is compared with various baseline techniques. Different DL-based approaches and other AI methods have been considered for the comparative analysis. The different models use different types of inputs for development. So, the performance evaluation in the comparative study with the baseline methods is conducted under uniform conditions, i.e., non-noisy conditions, and consistency is assured. The different approaches are based on AI techniques, like an artificial-neural-network-based model (ANN) and support vector machine (SVM) [45], a two-dimensional CNN (2D-CNN) [46], a combination of a CNN and SVM (CNN−SVM) [47], a Siamese convolutional autoencoder (SCA) [48], a combination of extreme machine learning and a CNN (EML−CNN) [49], a combination of ResNet and a convolutional block attention module (ResNet–CBAM) [50], a group normalization 2D CNN (GN-2D-CNN) [51], and a combination of a deep belief network and sparrow search algorithm (DBN−SSA) [52] methods. Table 4 displays the accuracy of the developed method, along with the other baseline methods. The table makes it clear that, when compared to different baseline approaches, the developed method has the highest accuracy. The techniques based on 2D-CNN [46] and GN-2D-CNN [51] have reasonable accuracy. These are 2D CNN models, and their training requires a great deal of labeled images as the input. However, the proposed RCNN model uses the 1D vibration data, which is advantageous. Also, the proposed methodology amalgamates the benefits of CNN and RNN, leading to efficient EV motor bearing fault detection.

Table 4. Performance comparison of the proposed method with various baseline methods.

Model	Accuracy (%)
ANN [45]	98.2
SVM [45]	92.5
2D-CNN [46]	99.13
CNN−SVM [47]	98.75
SCA [48]	96.41
EML−CNN [49]	99.2
ResNet–CBAM [50]	95.2
GN-2D-CNN [51]	99.37
DBN−SSA [52]	99.17
The Proposed Model (RCNN)	99.6

When it comes to detecting faults in the EV motor bearings, the RCNN model offers several benefits over the baseline techniques. This supremacy results from the ability of the model to combine the best features of RNNs and CNNs in a seamless manner. The ability of CNNs to extract spatial information from data, such as the local patterns within vibration signals, is essential to locate and diagnose defects at different sizes. RNNs, especially LSTM networks, excel at capturing temporal relationships, which makes them ideal for analyzing sequential data, like time-series vibration signals, where the dynamic behavior over time is critical. Combining these two algorithms enables the RCNN model to produce predictions that are more reliable and precise by tracking the evolution of complex spatial patterns over time, in addition to capturing them. In noisy environments with variable signal quality, such as varied SNR levels, this functionality is especially helpful as it may be difficult for baseline techniques like regular CNNs or SVMs to maintain accuracy. Furthermore, compared to the baseline techniques, the hierarchical structure layers of convolutions followed by recurrent layers of RCNNs enable superior generalization to an unknown input. The capacity of the model to learn both the local (spatial) and global (temporal) patterns that are pertinent across various data instances gives rise to this generalization ability, which also lessens the possibility of overfitting the training data, a problem that is frequently encountered by the baseline techniques. In essence, enhanced accuracy, robustness, and generalization result from the capacity of the RCNN model to capture and integrate both spatial and temporal features. This makes the model especially well-suited to challenging tasks like FD in electric vehicle motor bearings, where the spatial and temporal aspects of the data are critical for precise classification.

6. Conclusions

This paper presents an innovative EV motor bearing fault detection approach using the RCNN model. The proposed work combines two algorithms to develop an efficient FD strategy for EV electric motors. The RCNN architecture showed strong performance in categorizing various fault types by combining the advantages of LSTM layers for temporal sequence modeling and convolutional layers for feature extraction. By adding additive white Gaussian noise to the dataset, real-world scenarios were simulated, and the model was tested under noisy conditions. A variety of metrics, such as the confusion matrix, accuracy, precision, recall, F1-scores, and ROC curve, were employed to ensure a comprehensive assessment of the model's performance. The RCNN model showed constant superiority over the baseline techniques, especially when addressing the difficulties presented by a fluctuating SNR. The model's ability to discriminate between various fault classes was further demonstrated by the way its performance was visualized using ROC−AUC curves. Thereby, the RCNN model offers a dependable and effective method of discovering EV motor bearing faults in noisy environments, representing a substantial improvement in

the field of fault detection for EV motors. To further improve robustness, future studies could investigate integrating more noise-handling approaches and improvements to the model architecture.

Author Contributions: Conceptualization, P.K., P., A.K.S. and H.S.K.; methodology, P.K.; software, P.K. and P.; validation, P.K., P., A.K.S. and H.S.K.; formal analysis, P.K.; investigation, P.K.; resources, H.S.K.; data curation, P.K., P., and A.K.S.; writing—original draft preparation, P.K., P. and A.K.S.; writing—review and editing, P.K., P., A.K.S. and H.S.K.; supervision, H.S.K.; funding acquisition, H.S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Basic Science Research Program through a National Research Foundation of Korea grant (NRF-2020R1A2C1006613), funded by the Ministry of Education, and a National Research Foundation of Korea (NRF) grant (RS-2024-00405691), funded by the Korean government (MSIT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Lie, T.T.; Prasad, K.; Ding, N. The Electric Vehicle: A Review. *Int. J. Electr. Hybrid Veh.* **2017**, *9*, 49. [CrossRef]
2. El Hadraoui, H.; Zegrari, M.; Chebak, A.; Laayati, O.; Guennouni, N. A Multi-Criteria Analysis and Trends of Electric Motors for Electric Vehicles. *World Electr. Veh. J.* **2022**, *13*, 65. [CrossRef]
3. Krishnamoorthy, S.; Panikkar, P.P.K. A Comprehensive Review of Different Electric Motors for Electric Vehicles Application. *Int. J. Power Electron. Drive Syst. (IJPEDS)* **2024**, *15*, 74–90. [CrossRef]
4. Cai, W.; Wu, X.; Zhou, M.; Liang, Y.; Wang, Y. Review and Development of Electric Motor Systems and Electric Powertrains for New Energy Vehicles. *Automot. Innov.* **2021**, *4*, 3–22. [CrossRef]
5. Merve, Ş.K. Induction Motors—Recent Advances, New Perspectives and Applications. In *Electrical and Electronic Engineering*; IntechOpen: London, UK, 2023. [CrossRef]
6. Larminie, J.; Lowry, J. *Electric Vehicle Technology Explained*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2012; ISBN 978-1-118-36114-6.
7. Ehsani, M.; Gao, Y.; Emadi, A. *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2017; ISBN 978-1-315-21940-0.
8. Sinha, A.K.; Kumar, P.; Kumar, P. Electric Vehicle: Types, Charging Topologies, Energy Storage Management System and Applicability Micro-Grids-A Review. In *The Future of Hybrid-Electric Vehicles*; Nova Science Publishers, Inc.: Hauppauge, NY, USA, 2024; pp. 227–243.
9. Zhang, B.; Li, H.; Kong, W.; Fu, M.; Ma, J. Early-Stage Fault Diagnosis of Motor Bearing Based on Kurtosis Weighting and Fusion of Current–Vibration Signals. *Sensors* **2024**, *24*, 3373. [CrossRef]
10. Khaneghah, M.Z.; Alzayed, M.; Chaoui, H. Fault Detection and Diagnosis of the Electric Motor Drive and Battery System of Electric Vehicles. *Machines* **2023**, *11*, 713. [CrossRef]
11. Sinha, A.; Das, S.; Chatterjee, T. Wavelet Transform Based Ball Bearing Fault Detection Scheme for Heavy Duty Mining Electrical Motors under Supply Frequency Regulation Using MCSA. *Int. J. Technol.* **2018**, *9*, 170–180. [CrossRef]
12. Terron, C.; Martinez-Roman, J.; Puche-Panadero, R.; Sapena-Bañó, A. A Review of Techniques Used for Induction Machine Fault Modelling. *Sensors* **2021**, *21*, 4855. [CrossRef]
13. Mo, C.; Han, H.; Liu, M.; Zhang, Q.; Yang, T.; Zhang, F. Research on SVM-Based Bearing Fault Diagnosis Modeling and Multiple Swarm Genetic Algorithm Parameter Identification Method. *Mathematics* **2023**, *11*, 2864. [CrossRef]
14. Kumar, P.; Kumar, P.; Hati, A.S.; Kim, H.S. Deep Transfer Learning Framework for Bearing Fault Detection in Motors. *Mathematics* **2022**, *10*, 4683. [CrossRef]
15. Niu, G.; Dong, X.; Chen, Y. Motor Fault Diagnostics Based on Current Signatures: A Review. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3520919. [CrossRef]
16. Ghazali, M.; Rahiman, W. Vibration Analysis for Machine Monitoring and Diagnosis: A Systematic Review. *Shock Vib.* **2021**, *2021*, 9469318. [CrossRef]
17. Yang, Y.; Haque, M.M.M.; Bai, D.; Tang, W. Fault Diagnosis of Electric Motors Using Deep Learning Algorithms and Its Application: A Review. *Energies* **2021**, *14*, 7017. [CrossRef]
18. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]

19. Kumar, P.; Hati, A.S. Dilated Convolutional Neural Network Based Model for Bearing Faults and Broken Rotor Bar Detection in Squirrel Cage Induction Motors. *Expert Syst. Appl.* **2022**, *191*, 116290. [CrossRef]
20. Kumar, P.; Hati, A.S. Deep Convolutional Neural Network Based on Adaptive Gradient Optimizer for Fault Detection in SCIM. *ISA Trans.* **2021**, *111*, 350–359. [CrossRef] [PubMed]
21. Zhao, D.; Cai, W.; Cui, L. Adaptive Thresholding and Coordinate Attention-Based Tree-Inspired Network for Aero-Engine Bearing Health Monitoring under Strong Noise. *Adv. Eng. Inform.* **2024**, *61*, 102559. [CrossRef]
22. Kumar, P.; Raouf, I.; Kim, H.S. Transfer Learning for Servomotor Bearing Fault Detection in the Industrial Robot. *Adv. Eng. Softw.* **2024**, *194*, 103672. [CrossRef]
23. Zhao, D.; Shao, D.; Cui, L. CTNet: A Data-Driven Time-Frequency Technique for Wind Turbines Fault Diagnosis under Time-Varying Speeds. *ISA Trans.* **2024**, in press. [CrossRef]
24. Raouf, I.; Kumar, P.; Soo Kim, H. Deep Learning-Based Fault Diagnosis of Servo Motor Bearing Using the Attention-Guided Feature Aggregation Network. *Expert Syst. Appl.* **2024**, *258*, 125137. [CrossRef]
25. Ince, T.; Kiranyaz, S.; Gabbouj, M. A Generic and Robust System for Automated Patient-Specific Classification of ECG Signals. *IEEE Trans. Biomed. Eng.* **2009**, *56*, 1415–1426. [CrossRef] [PubMed]
26. Tolic, A.; Boshkoska, B.M.; Skansi, S. Chrono Initialized LSTM Networks with Layer Normalization. *IEEE Access* **2024**, *12*, 115219–115236. [CrossRef]
27. Li, X.; Chang, H.; Wei, R.; Huang, S.; Chen, S.; He, Z.; Ouyang, D. Online Prediction of Electric Vehicle Battery Failure Using LSTM Network. *Energies* **2023**, *16*, 4733. [CrossRef]
28. Xu, L.; Teoh, S.S.; Ibrahim, H. A Deep Learning Approach for Electric Motor Fault Diagnosis Based on Modified InceptionV3. *Sci. Rep.* **2024**, *14*, 12344. [CrossRef]
29. He, M.; He, D. Deep Learning Based Approach for Bearing Fault Diagnosis. *IEEE Trans. Ind. Appl.* **2017**, *53*, 3057–3065. [CrossRef]
30. Li, Y.; Wang, Y.; Wang, R.; Wang, Y.; Wang, K.; Wang, X.; Cao, W.; Xiang, W. Ga-Cnn: Convolutional Neural Network Based on Geometric Algebra for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5540314. [CrossRef]
31. Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent Advances in Recurrent Neural Networks. *arXiv* **2017**, arXiv:1801.01078.
32. Suthers, D.D.; Dwyer, N.; Medina, R.; Vatrapu, R. A Framework for Conceptualizing, Representing, and Analyzing Distributed Interaction. *Int. J. Comput. Support. Collab. Learn.* **2010**, *5*, 5–42. [CrossRef]
33. Le, M.D.; Rathour, V.S.; Truong, Q.S.; Mai, Q.; Brijesh, P.; Le, N. Multi-Module Recurrent Convolutional Neural Network with Transformer Encoder for ECG Arrhythmia Classification. In Proceedings of the 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), Athens, Greece, 27–30 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [CrossRef]
35. Tsai, M.-F.; Tseng, H.-J. Enhancing the Identification Accuracy of Deep Learning Object Detection Using Natural Language Processing. *J. Supercomput.* **2021**, *77*, 6676–6691. [CrossRef]
36. Zhao, J.; Deng, F.; Cai, Y.; Chen, J. Long Short-Term Memory-Fully Connected (LSTM-FC) Neural Network for PM2. 5 Concentration Prediction. *Chemosphere* **2019**, *220*, 486–492. [CrossRef] [PubMed]
37. Yıldırım, Ö.; Baloglu, U.B.; Acharya, U.R. A Deep Convolutional Neural Network Model for Automated Identification of Abnormal EEG Signals. *Neural Comput. Appl.* **2020**, *32*, 15857–15868. [CrossRef]
38. Hati, A.S. Convolutional Neural Network-Long Short Term Memory Optimization for Accurate Prediction of Airflow in a Ventilation System. *Expert Syst. Appl.* **2022**, *195*, 116618.
39. Vu, N.T.; Adel, H.; Gupta, P.; Schütze, H. Combining Recurrent and Convolutional Neural Networks for Relation Classification. *arXiv* **2016**, arXiv:1605.07333.
40. Wang, R.; Li, Z.; Cao, J.; Chen, T.; Wang, L. Convolutional Recurrent Neural Networks for Text Classification. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
41. Xie, X.; Wang, B.; Wan, T.; Tang, W. Multivariate Abnormal Detection for Industrial Control Systems Using 1D CNN and GRU. *IEEE Access* **2020**, *8*, 88348–88359. [CrossRef]
42. Kumar, P.; Shankar Hati, A. Convolutional Neural Network with Batch Normalisation for Fault Detection in Squirrel Cage Induction Motor. *IET Electr. Power Appl.* **2021**, *15*, 39–50. [CrossRef]
43. Loparo, K.A. Case Western Reserve University Bearing Data Center. In *Bearings Vibration Data Sets*; Case Western Reserve University: Cleveland, OH, USA, 2012; pp. 22–28.
44. He, Q.; Li, S.; Bai, Q.; Zhang, A.; Yang, J.; Shen, M. A Siamese Vision Transformer for Bearings Fault Diagnosis. *Micromachines* **2022**, *13*, 1656. [CrossRef]
45. Sharma, A.; Mathew, L.; Chatterji, S.; Goyal, D. Artificial Intelligence-Based Fault Diagnosis for Condition Monitoring of Electric Motors. *Int. J. Patt. Recogn. Artif. Intell.* **2020**, *34*, 2059043. [CrossRef]
46. Magar, R.; Ghule, L.; Li, J.; Zhao, Y.; Farimani, A.B. FaultNet: A Deep Convolutional Neural Network for Bearing Fault Classification. *IEEE Access* **2021**, *9*, 25189–25199. [CrossRef]
47. Yuan, L.; Lian, D.; Kang, X.; Chen, Y.; Zhai, K. Rolling Bearing Fault Diagnosis Based on Convolutional Neural Network and Support Vector Machine. *IEEE Access* **2020**, *8*, 137395–137406. [CrossRef]

48. Hu, H.-X.; Cao, C.; Hu, Q.; Zhang, Y.; Lin, Z.-Z. A Real-Time Bearing Fault Diagnosis Model Based on Siamese Convolutional Autoencoder in Industrial Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 3820–3831. [CrossRef]
49. Zhang, Y.; Shang, L.; Gao, H.; He, Y.; Xu, X.; Chen, Y. A New Method for Diagnosing Motor Bearing Faults Based on Gramian Angular Field Image Coding and Improved CNN-ELM. *IEEE Access* **2023**, *11*, 11337–11349. [CrossRef]
50. Liang, B.; Feng, W. Bearing Fault Diagnosis Based on ICEEMDAN Deep Learning Network. *Processes* **2023**, *11*, 2440. [CrossRef]
51. Gilbert Chandra, D.; Srinivasulu Reddy, U.; Uma, G.; Umapathy, M. Group Normalization-Based 2D-Convolutional Neural Network for Intelligent Bearing Fault Diagnosis. *J. Braz. Soc. Mech. Sci. Eng.* **2023**, *45*, 584. [CrossRef]
52. Xu, D.; Li, C. Optimization of Deep Belief Network Based on Sparrow Search Algorithm for Rolling Bearing Fault Diagnosis. *IEEE Access* **2024**, *12*, 10470–10481. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.