IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 March 2024, accepted 11 April 2024, date of publication 17 April 2024, date of current version 1 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3390234

RESEARCH ARTICLE

A Novel Framework for Robust Bearing Fault Diagnosis: Preprocessing, Model Selection, and Performance Evaluation

FAISAL ALTHOBIANI
Marine Engineering Department, Faculty of Maritime Studies, King Abdulaziz University, Jeddah 21589, Saudi Arabia
e-mail: falthobiani@kau.edu.sa

This research work was funded by Institutional Fund Projects under grant no. (IFPIP-450-980-1442). Therefore, the authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, Deanship of Scientific Research (DSR), Jeddah, Saudi Arabia.

ABSTRACT Diagnosing bearing faults is crucial for maintaining, ensuring reliability, and extending the lifespan of rotary machines. This process helps prevent unexpected downtime in industries, ultimately reducing economic losses caused by the failure of rotary machines. Timely diagnosis of bearing faults is crucial to prevent catastrophic breakdowns, minimize maintenance expenses, and ensure uninterrupted productivity. With industries evolving rapidly and machines operating in increasingly diverse conditions, traditional fault detection methods face limitations. Despite extensive research in recent decades, there is an ongoing need for further advancements to enhance existing fault diagnosis techniques. This study addresses these challenges by utilizing advanced machine learning algorithms Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Gated Recurrent Unit Network (GRU), Bidirectional LSTM, and for precise bearing fault diagnosis. Leveraging the CWRU dataset encompassing diverse fault classes and machine conditions, a comprehensive data preprocessing pipeline was executed to clean, normalize, and augment the dataset, ensuring model readiness and enhancing performance. Performance analysis revealed the proposed models achieving remarkable accuracies on the CWRU dataset. The CNN and LSTM models attained accuracies of 95%, while the RNN and GRU models achieved accuracies of 97%. Additionally, the Bidirectional LSTM model yielded an accuracy of 96%. These results signify substantial advancements in bearing fault diagnosis, emphasizing the models' efficacy in accurately detecting and categorizing faults within the 10 classes of the CWRU dataset. The findings underscore the potential of advanced machine learning techniques in revolutionizing fault diagnosis for rotary machines, addressing the persistent need for more robust and accurate diagnostic methodologies.

INDEX TERMS Bearing fault, deep learning, intelligent solution, reliability, protection.

I. INTRODUCTION

Electrical devices are being used significantly in several industrial uses. As the manufacturing of applications is growing rapidly, machines are now working in unfavourable conditions on daily basis. This condition leads to many flaws in machinery. Such conditions also subject the machine to extreme dampness and overloading situations, which ultimately cause failure of machinery [1]. When the fault is not identified early on stage, It may have disastrous effects in terms of adverse maintenance duration and expense, a drop in productivity [2]. Each rotatory machinery holds a bearing because its primary function is to minimize the friction and sustain the load. The bearing is comprising of four essential component parts: the cage, the inner race, the outer race, and balls. The cage effectively holds the balls within position between the races inside and outside, facilitating unhindered and seamless rotation [3]. An impulsive force is generated when an error on the one bearing's area collides with the area of another bearing. This defect has been capitalized by various techniques for vibration analysis and signal processing techniques. Bearing faults are most

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

prevalent types of faults accounting of almost 30 percent of all faults in the context of induction machinery defects. Furthermore, a significant loss of assets and security can be caused because bearing fault is a typical cause of the motor shutting off. Pertaining these mentioned factors, bearing fault diagnosis holds significant importance in both development and engineering [4].

As the signal provides dynamic data regarding the condition of the device, fault detection primarily depends on feature extraction techniques. Specific features are exhibited by vibration signal patterns as a result of defect in several rotating components. Analysis of pattern irregularities helps in the detection of anomalies in rotatable components and plant machinery. In the similar context, several research have discussed signal processing techniques like the three domains are time, frequency, and combined time-frequency (similar to wavelet transformation). The signals of vibration produced by devices are frequently categorised as uncertainty because of differences in friction, loading conditions, interactions between various rotating components, bearing clearance as well as nonlinear rigidity. Therefore, several researchers have focused on nonlinear parameter estimation techniques [5].

Certain methods of fault diagnosis such as motor current signature analysis (MCSA), stray flux monitoring (SFM) and acoustic emission (AE) are commonly practised. Vibration analysis is traditional and trustworthy method, however, certain issues such as placement of vibration sensors, cost and access makes it difficult to achieve practical application. The Motor Current Signature Analysis (MCSA) offers a non-invasive alternative to the traditional vibration analysis system. The primary benefit of MCSA is that it doesn't need any access to specific sensors or bearing. However, MCSA still has limitations as it does not provide a precise assessment of the existing situation within the bearing condition [6], [7]. There is a need of direct access to motor for AE and stray flux methods. Numerous studies are currently underway to discover the most optimal and efficient a method for identifying bearing failure. Non-invasive methods offer cost effectiveness and easy accessibility. Therefore, the development of an efficient condition monitoring system using a non-invasive method becomes imperative for early detection of faults [8].

In the past few decades, there has been an extension in digital signal processing for the purpose of identifying bearing issues in induction motors. It interprets the statistical data and generate reliable output. It is impossible to find errors with the raw signals obtained from the experiment [9]. There are three method such as Analysis in the time, frequency, or time-frequency domains that can be employed to remove the characteristics. Furthermore, several research have also considered hole as a faulty factor for the analysis if bearing failure. Similarly, scratches on bearings have a relatively high probability of occurrence and must be taken into account when conducting bearing failure analysis. Early detection of faults, particularly during the minor stage, is crucial to reducing sudden breakdowns in industrial settings. Therefore, research focusing on the detection of scratches and minor faults deserves significant attention [10].

Bearing fault diagnosis depends on the use of Artificial Intelligence (AI) and Machine Learning (ML) to improve the Condition Monitoring (CM) system, ultimately leading to improved motor reliability. The application of ML gets globalized as it is being used in market analysis, telecommunication, weather prediction, image sensing, medical diagnosis etc [11]. In order to detect the bearing fault (BF), the traditional methods solely relied on principles at specific rates of failure to identify the fault. However, there are numerous concealed and distinctive interactions within the information that could indicate the presence of a bearing failure. It might be impossible for a human to understand these relationships. Therefore, machine learning algorithms can be implemented to identify the machine's malfunction. These creative approaches involve analysing the data, training on it, and learning from it. Once these approaches are trained and have enough knowledge, they use this data to make the final choice regarding the existence of the bearing fault [12]. Therefore, it is highly advantageous to utilize artificial intelligence methods for creating a new system based on knowledge. It is possible to detect bearing faults at early stages which helps prevent significant equipment damage and reduces overall operating expenses [13].

Bearing fault diagnosis has always remained a challenging task. Therefore, it has always been an area of interest for researchers. Several researchers have focused on applying machine learning methods such as K nearest neighbour (KNN), Artificial Neural Network (ANN), Deep Learning (DL), Support Vector Machine (SVM), Decision Tree (DT) etc [14]. According to Li [15], fault detection is considered as classification problem where it is treated by categorizing different types of machine faults. Features are derived off of the machine's signals. The authors focused on the vibration acceleration signals because they can be collected easily and have been extensively used in industries. However, this also becomes the limitation of this study because it neglects the other types of signals in machines. In recent decade, deep learning model is considered as most effective model for pattern recognition. It assists in overcoming the challenges in current fault diagnosis system. The architecture of deep learning shows a stack of network where information from raw data can be collected. However, it is technically expensive to apply this technique because the model contains several numbers of layers [16].

Similarly, the authors of [17] utilized Convolution Neural Network (CNN) which is a particular kind of deep learning framework and Particularly developed for complex signals. CNN is widely used in processing bearing fault signals and has proven as an efficient approach for extracting features. However, the result of this model Highly relies on the value of the signal. This means that noisy and uneven data can have a significant effect on the outcomes of the model.

Guo et al. [18] introduced a learning rate hierarchy deep convolutional neural network with adaptability for diagnosing bearing flaws and assessing their severity. The network consists of three stacked layers of pooling and convolution below the top fully connected classifier. However, finding the optimal architecture and hyperparameter settings can be challenging.

In the similar context, Li et al. [19], stated a method known as categorization of multi-modal deep support vectors using the addition of similar function. In order to diagnose specific health of the machinery, authors utilized stack denoises auto encoder. They applied this method to signals that included ambient noise and fluctuations in working conditions. The results from the experiments demonstrate that the suggested deep learning technique based on separation-fusion works well for diagnosing faults in machinery. The complexity of the proposed model makes the interpretability difficult to attain in results. In the related study, Zhang et al. [20] utilized an adaptive batch normalization technique to enhance the neural network's capability to adapt to different domains. Furthermore, the study also showed that the results can be effectively applied and adapted to various operating scenarios in different types of machinery. The outcomes indicate that deep learning is effective in extracting domain-invariant features for bearing diagnosis. Although the study focuses on the diagnosis of bearing faut, however, the performance and usefulness of the model still needs to be studied. A pioneering study from Jia et al. [21] visualized the kernels of the convolutional layers in bearing fault diagnosis, providing valuable insights into how neural networks perform their convolutional operations effectively. However, the study focuses on the visualization of convolution layer only. There are several types of other layers such as pooling layer that assist in diagnosis system.

Current bearing fault diagnosis techniques, including vibration analysis and traditional signal processing methods, have several limitations. Firstly, these techniques often rely on manual inspection or periodic monitoring, which can result in missed or delayed detection of faults. Additionally, they struggle to effectively differentiate between normal operating conditions and early-stage fault signatures, leading to false alarms or overlooked issues. Furthermore, the complexity of machinery dynamics and the presence of background noise pose significant challenges in accurately identifying subtle fault patterns. To address these limitations, this study proposes a novel approach leveraging deep learning techniques for bearing fault diagnosis. Deep learning offers several advantages over traditional methods, including its ability to automatically learn relevant features from raw sensor data, its capacity to handle complex nonlinear relationships, and its potential for continuous learning and adaptation to changing operating conditions. This paper presents a comprehensive analysis of multiple deep learning architectures, including Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), Long Short-Term Memory networks (LSTMs), Bidirectional LSTM, and Recurrent Neural Networks (RNNs), for bearing fault diagnosis. By harnessing the power of deep learning, I aim to overcome the limitations of current techniques and enhance the accuracy and efficiency of fault detection in industrial machinery. This research contributes to the existing body of knowledge by offering a systematic evaluation of deep learning models for bearing fault diagnosis, highlighting their strengths and limitations compared to traditional methods.

The remaining paper is organized as follows: Section II describes the materials and methods applied for the proposed work. Section III explains the results and discussion, and finally, the paper is concluded in the conclusion section.

II. MATERIALS AND METHODOLOGY

A. DATASET DESCRIPTION

The dataset utilized in this study originates from the Case Western Bearing Data Center, representing a collective contribution aimed at providing motor performance data available on Kaggle. It comprises comprehensive information concerning the assessment of motor performance, encompassing various components:

- Equipment: The test bench employed for motor performance evaluation involves a motor generating 2 horsepower, complemented by a torque transducer, dynamometer, and control electronics.
- Test Bearings: Defects were intentionally induced at a single point within the test bearings using Electrical Discharge Machining (EDM). The sizes of these defects are measured in inches (millimeters) and include diameters of 0.007 inches (0.178 millimeters), 0.014 inches (0.356 millimeters), and 0.021 inches (0.533 millimeters).
- Time Series and Bearing Parts: Each defect is associated with a time series, specifically located in one of three segments of the bearing: ball, inner race, or outer race.
- Telemetry Measurements: The data includes telemetry readings from three accelerometers strategically positioned in the system: Drive end (DE), Fan end (FE), and Base (BA). Experimental Conditions: The dataset corresponds to a specific set of conditions:
- Load: A 1 HP load applied to the motor.
- Rotational Speed: Motor shaft rotating at a speed of 1772 rotations per minute (rpm).
- Sampling Frequency: Accelerometer data was sampled at a frequency of 48 kHz.

Extracted Features: To facilitate fault identification and prediction, nine essential features were derived from the data. These features were calculated based on segments of 2048 data points, corresponding to a duration of 0.04 seconds at the 48 kHz accelerometer sampling frequency. The computed features include:

$$\mathbf{max} = \mathbf{max}(\mathbf{data}) \tag{1}$$

$$\mathbf{min} = \mathbf{min}(\mathbf{data}) \tag{2}$$

$$\mathbf{mean} = \frac{\mathbf{sum}(\mathbf{data})}{\mathbf{n}} \tag{3}$$

$$\mathbf{sd} = \sqrt{\frac{1}{\mathbf{n}}\sum_{i=1}^{\mathbf{n}}(\mathbf{data}_i - \mathbf{mean})^2} \tag{4}$$

$$\mathbf{rms} = \sqrt{\frac{1}{\mathbf{n}}\sum_{i=1}^{\mathbf{n}}\left(\mathbf{data}_i^2\right)} \tag{5}$$

$$\mathbf{skewness} = \frac{\frac{1}{\mathbf{n}}\sum_{i=1}^{\mathbf{n}}(\mathbf{data}_i - \mathbf{mean})^3}{\mathbf{sd}^3} \tag{6}$$

$$\mathbf{kurtosis} = \frac{\frac{1}{\mathbf{n}}\sum_{i=1}^{\mathbf{n}}(\mathbf{data}_i - \mathbf{mean})^4}{\mathbf{sd}^4} \tag{7}$$

$$\mathbf{Crest} = \frac{\mathbf{max}}{\mathbf{rms}} \tag{8}$$

$$\mathbf{Form} = \frac{\mathbf{rms}}{\mathbf{mean}} \tag{9}$$

Following is the detail of parameters: Maximum or max encapsulate all the highest recorded sensor value within the dataset. Whereas its counterpart is Minimum or min which records the lowest sensor value of the dataset. Mean parameter shows the average of the sensor reading and offer measure of central tendency. Root mean square provides a sense of the amplitude of the data by defining the total magnitude of the readings. Skewness explains the form of the dataset by reflecting the asymmetry of the data distribution. Similarly, kurtosis reveals the existence of outliers or extreme values by displaying the form of the data's distribution. The crest factor is the ratio of the peak value of the waveform to its rms (Root Mean Square) value. The form factor is the ratio of rms value to the average value of the waveform.

The provided dataset encompasses a range of class labels that delineate distinct fault types and conditions within the bearing system. Each class label denotes specific fault configurations, encompassing varying defect sizes and locations across different parts of the bearing assembly. Below are detailed descriptions of each class label and its corresponding fault type within the system:

- Ball_007_1: Refers to a fault in the ball bearing with a diameter defect of 0.007 inches (0.178 millimeters). This class represents the specific type of fault in the ball bearing with the mentioned defect size.
- Ball_014_1: Indicates a fault in the ball bearing with a diameter defect of 0.014 inches (0.356 millimeters). Similar to the previous class, this represents a different defect size within the ball bearing.
- Ball_021_1: Represents a fault in the ball bearing with a diameter defect of 0.021 inches (0.533 millimeters). This class signifies yet another defect size within the ball bearing category.
- IR_007_1: Denotes a fault in the inner race of the bearing with a diameter defect of 0.007 inches (0.178 millimeters). This class pertains specifically to faults occurring in the inner race of the bearing with the mentioned defect size.
- IR_014_1: Refers to a fault in the inner race of the bearing with a diameter defect of 0.014 inches (0.356 millimeters). Similar to the previous class, this represents a different defect size within the inner race category.
- IR_021_1: Signifies a fault in the inner race of the bearing with a diameter defect of 0.021 inches (0.533 millimeters). This class represents yet another defect size within the inner race category.
- Normal_1: Represents the normal or healthy state of the system without any introduced faults or defects. This class serves as the baseline or reference for normal operating conditions.
- OR_007_6_1: Denotes a fault in the outer race of the bearing with a diameter defect of 0.007 inches (0.178 millimeters). This class pertains specifically to faults occurring in the outer race of the bearing with the mentioned defect size.
- OR_014_6_1: Represents a fault in the outer race of the bearing with a diameter defect of 0.014 inches (0.356 millimeters). Similar to the previous class, this represents a different defect size within the outer race category.
- OR_021_6_1: Signifies a fault in the outer race of the bearing with a diameter defect of 0.021 inches (0.533 millimeters). This class represents yet another defect size within the outer race category.

B. DATASET PREPROCESSING

Effective data preprocessing is a critical phase in preparing raw datasets for machine learning models. In the context of our study, several essential preprocessing steps were performed to refine and optimize the dataset before model training and testing.

1) FEATURE EXTRACTION AND TARGET DEFINITION

The initial dataset was structured to encapsulate a range of features relevant to motor performance evaluation. Features such as maximum, minimum, mean, standard deviation, RMS, skewness, kurtosis, crest factor, and form factor were extracted to characterize distinct time segments within the data. Simultaneously, class labels representing various fault types and normal operational conditions were defined, forming the target variable for classification.

2) DATA SEGREGATION

The features and target labels were segregated into separate entities, with features stored in X and target labels in y.

3) ENCODING CATEGORICAL LABELS

To facilitate model training, categorical class labels were encoded into numerical representations using LabelEncoder, ensuring compatibility with machine learning algorithms.

4) TRAIN-TEST SPLIT

The dataset was partitioned into training and testing sets using train_test_split. This separation allowed for the assessment of model performance on unseen data, with 85

5) FEATURE SCALING

Standardization of feature values was executed through StandardScaler, ensuring uniformity by centering features around zero mean and scaling to unit variance. This step aids in preventing certain features from disproportionately influencing model training.

6) DATA RESHAPING

Lastly, to accommodate specific model requirements, the feature data was reshaped by adding an additional axis, converting the shape to a format compatible with models expecting three-dimensional inputs.

These preprocessing steps collectively refine the dataset, ensuring that features are appropriately scaled, encoded, and structured for optimal utilization in training machine learning models to predict and classify fault types within the motor performance assessment domain.

C. PROPOSED ALGORITHMS

1) CONVOLUTION NEURAL NETWORK (CNN)

In image processing and natural language processing, convolutional neural networks (CNNs) are a type of neural network with convolution that is subsequent computation as well as significant structure. Convolutional and pooling layers are present in each CNN hidden layer. The convolutional layer maps the nearby signal of the preceding layer to the following layer using a shared filter weights to extract characteristics from the signal received. Shared weights provide substantial advantages and significantly lower the processing burden for difficult nonlinear transformations. The vibration signal in this study is a one-dimensional time series, therefore one-dimensional convolution was applied.

$$y_i^{i+1}(j) = k_i^j \times x^i(j) + b_i^j \tag{10}$$

This equation determines every single neuron's output in the layer following it. It integrates the input from the preceding layer with the weight and bias related to that neuron. This process is repeated for each neuron in the layer under consideration. The result of a layer of a neural network is represented by this equation. For a particular neuron (i) in that layer, it determines the value for the following layer (l+1). It's determined by a weight (K) and bias (b) associated with the l-th layer and the i-th filter. The input value (xl) from the previous layer (j) is multiplied by both bias and weight to determine the value [22].

$$P_i^{i+\mathbf{1}}(j) = \frac{max}{(j-\mathbf{1})W + 1 \le t \le jW} \left\{ q_i^l(t) \right\} \tag{11}$$

A pooling procedure is explained by the second equation. In a neural network, we often need to minimise the quantity of data after extracting features via convolution. Max-pooling is a common method where, for each region, the largest value is selected. The equation demonstrates how this operation is applied to obtain values for the next layer. The equation describes the outcome of a pooling procedure, in which we choose certain values from a set. It specifically describes the outcome for the i-th channel in the (l+1)-th layer following pooling. The highest value is chosen from a range of values in the preceding layer, from (j1) W+1 to jW. This is carried out for every neuron in the l-th layer (i(t)), where 'w' stands for the width of the kernel for pooling [23].

TABLE 1. CNN model summary.

Layer (type)	Output Shape	Param #
Conv1D	(None, 9, 16)	32
MaxPooling1D	(None, 4, 16)	0
Flatten	(None, 64)	0
Dense	(None, 128)	8,320
Dense	(None, 64)	8,256
Dense	(None, 10)	650
Total params		**17,258**
Trainable params		**17,258**
Non-trainable params		**0**

Table 1 provides a summary of the architecture for CNN model. The model was trained using the Adam optimizer, with a sparse categorical cross-entropy loss function and softmax activation function.

D. LONG SHORT-TERM MEMORY

Predictive modelling challenges involving prediction of time series are complex. The issues of time series prediction in predictive modelling are complicated. Time series, as opposed to regression predictive modelling, also adds complexity by establishing a connection between the sequences of the input variables. A strong relations of neural networks developed to manage a series dependency are recurrent neural networks. The Long Short-Term Memory network, often known as the LSTM network, is a form of RNN utilised in machine learning because very complex designs may be learned. It was trained via backpropagation over time. It is used to construct large recurrent networks in order to solve complex sequence problems in machine learning and deliver cutting-edge results. In contrast to neurons, memory blocks in LSTM networks are linked by layers.

$$it = \sigma(wi \times ht - \mathbf{1} + xt + bi) \tag{12}$$

$$ft = \sigma(wf \times ht - \mathbf{1} + xt + bf) \tag{13}$$

$$ot = \sigma(wo \times ht - \mathbf{1} + xt + bo) \tag{14}$$

LSTM networks are a distinct subclass of preferred recurrent neural networks at handling sequential input. They do this by including a special gating device that regulates the information flow inside the network. The idea of LSTM was presented in 1997 by Hochreiter and Schmidhuber, aiming to improve the modelling of Long-range connections between temporal sequences, a task often challenging for conventional Recurrent Neural Networks (RNNs). The three main gates that are utilized are the input gate, the forget gate, and the output gate make up each LSTM unit. These gates serve as filters, selecting the data that should be left to enter the

TABLE 2. LSTM model summary.

Layer (type)	Output Shape	Param #
LSTM	(None, 9, 64)	16,896
LSTM	(None, 9, 64)	33,024
Dropout	(None, 9, 64)	0
Flatten	(None, 576)	0
Dense	(None, 128)	73,856
Dense	(None, 64)	8,256
Dropout	(None, 64)	0
Dense	(None, 10)	650
Total params		**132,682**
Trainable params		**132,682**
Non-trainable params		**0**

cell. An activation function sigmoid, which generates values between zero and one and effectively functions as an on-off switch, controls how they operate. Information passes through a gate when it is open (value near one) and is stopped when it is closed (value near zero) [24].

Particularly, ''it'' stands for the input gate, ''ft'' for the forget gate, and ''ot'' for the output gate. The numbers are kept within the range of zero and one due to the sigmoid function. ''ht-1'' is the output from the prior Block of LSTM at time-step ''t-1,'' ''xt'' is the input at the present time-step, and ''wi,'' ''wf,'' and ''wo'' are the weights for the corresponding gates. ''bi,'' ''bf,'' and ''bo'' are the biases for the corresponding gates [25].

Table 2 provides a summary of the architecture for LSTM model. The model was trained using the Adam optimizer, with a sparse categorical cross-entropy loss function and softmax activation function.

E. RECURRENT NEURAL NETWORK

A type of artificial neural network called a recurrent neural network (RNN) is one where relationships among nodes create a focused loop. With this design, the network can maintain a state over multiple time step. Recurrent neural networks (RNNs) are a subset of artificial neural networks that are specifically designed to process sequential data. Because of their proficiency in processing sequences, they are especially well-suited for a variety of tasks involving ordered data, including natural language processing, speech recognition, time series analysis, and other sequential applications. The use of feedback loops is a crucial idea that supports the operation of the RNN. The purpose of these feedback loops is to gather and store data from previous network states, which they then use to influence predictions and output generation. RNNs are an essential tool for tasks that need temporal dependencies and context preservation because of their iterative process, which enables them to demonstrate a unique capacity to recall and take into consideration the context of previous parts within a sequence [28]. The input layer of a neural network, denoted by the letter ''x,'' is where

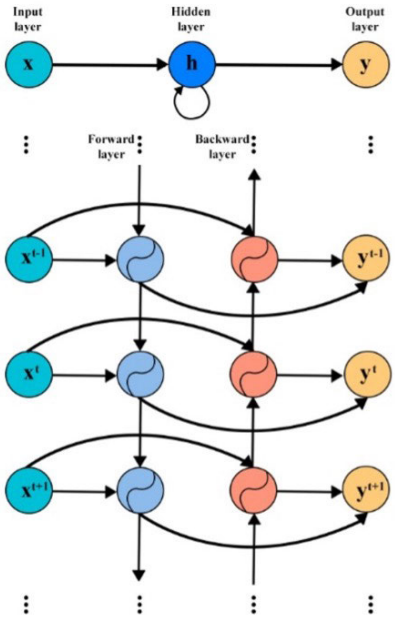

FIGURE 1. Structure of recurrent neural network.

TABLE 3. RNN model summary.

Layer (type)	Output Shape	Param #
SimpleRNN	(None, 64)	4,224
Dense	(None, 10)	650
Total params		**4,874**
Trainable params		**4,874**
Non-trainable params		**0**

the first data enters the network and is processed before being sent to the middle layer.

Several hidden layers, each having unique weights, biases, and activation functions, may be included in the intermediate layer, generally represented by the letter ''h.'' A recurrent neural network is a possibility if working with a neural network in which the hidden layer parameters don't depend on the preceding layer, i.e., the network has no memory. By establishing the activation algorithms and parameters such that all hidden layers have the same values, a recurrent neural network makes setup easier. Rather than generating several discrete hidden layers, it combines them into a single layer and iterates the procedure as frequently as required. As a result, the network becomes more efficient and can handle positions that don't need reference or memory maintenance.

Table 3 provides a summary of the architecture for RNN model. The model was trained using the Adam optimizer, with a sparse categorical cross-entropy loss function and softmax activation function.

F. GATED RECURRENT UNIT NETWORKS

GRU is made to handle subsequent data, like voice, text, and time-series data, much like LSTMs. GRU's fundamental

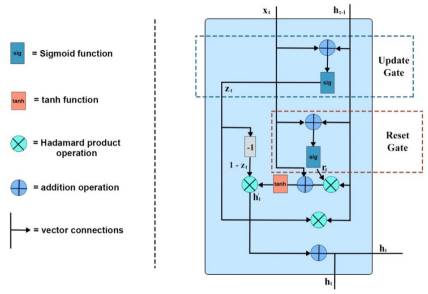

FIGURE 2. Structure of gated recurrent unit networks.

TABLE 4. GRU model summary.

Layer (type)	Output Shape	Param #
GRU	(None, 64)	12,864
Dense	(None, 128)	8,320
Dense	(None, 10)	1,290
Total params		**22,474**
Trainable params		**22,474**
Non-trainable params		**0**

idea is the application of gating mechanisms, which are essential in figuring out how the hidden state of the network changes with each passing time step. By acting as controllers, these gating mechanisms control the information entering and departing the network. The reset gate and the update gate, two essential gating processes, are integrated into the GRU. The update gate determines how much of the new data has to be utilised to change the hidden state, while the reset gate controls how much the previously hidden state ought to be forgotten by the network. This updated hidden state is employed in order to compute the GRU's output, ensuring that relevant data is collected, and memory and context are managed effectively [29].

Table 4 provides a summary of the architecture for GRU model. The model was trained using the Adam optimizer, with a sparse categorical cross-entropy loss function and softmax activation function.

G. BI-DIRECTIONAL MODEL

Bidirectional LSTM is used in sequence modelling. Two LSTM layers are used in this process: one for forward data processing and another for backward data processing. This method is frequently used in tasks including natural language processing. This strategy aims to improve the model's comprehension of the connections between sequences, for example, by making it more aware of the words that come before and after a given word in a phrase. Two distinct unidirectional LSTM networks, one for processing the sequence in its original order and the other for processing it in reverse, make up the design of a bidirectional LSTM. The output of each of these LSTM networks is a probability vector. The information from these two probability vectors

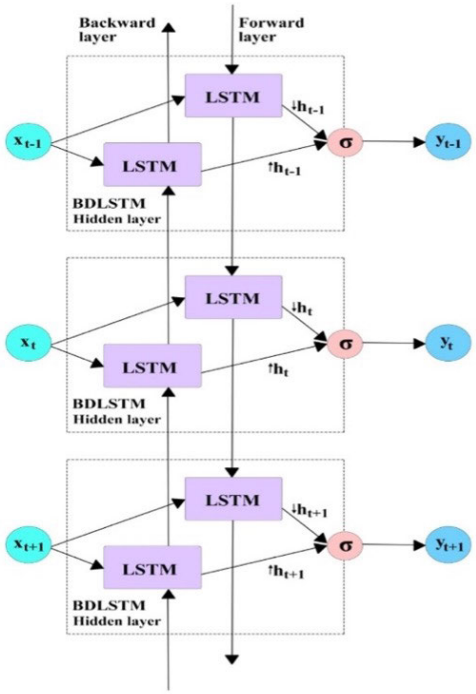

FIGURE 3. Structure of Bi-directional LSTM model.

TABLE 5. Bi-directional LSTM model summary.

Layer (type)	Output Shape	Param #
Bidirectional	(None, 9, 256)	133,120
Dropout	(None, 9, 256)	0
Bidirectional	(None, 128)	164,352
Dropout	(None, 128)	0
Dense	(None, 64)	8,256
Dense	(None, 10)	650
Total params		**306,378**
Trainable params		**306,378**
Non-trainable params		**0**

is combined in the model's final output, which enables the model to capture a more complete picture of the sequential data [31].

Table 5 provides a summary of the architecture for Bidirectional model. The model was trained using the Adam optimizer, with a sparse categorical cross-entropy loss function and softmax activation function.

III. RESULTS AND DISCUSSIONS

The proposed model evaluates algorithm performance using metrics such as precision, recall, F1 score, and accuracy. Their mathematical notations are presented in equations (15)-(18). F1 score, a crucial machine learning parameter, provides insights into algorithm accuracy by combining precision and recall performance. It gauges how accurately the model predicts results across the entire dataset.

It's important to note that achieving reliable F1 score results requires a balanced dataset. Precision, another evaluation metric, measures the proportion of accurate positive class predictions, indicating how many positive predictions were correct. Recall evaluates the model's ability to identify positive instances by dividing correctly classified positive samples by the total number of positive samples. A higher recall implies a better ability to detect positive samples. Similarly, accuracy is an evaluation metric that calculates the number of accurate predictions made by the algorithm in relation to the total predictions. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the four numbers that represent the quantity of positives.

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{16}$$

$$Specificity = \frac{TN}{TN + FP} \tag{17}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

A. UNDERSTANDING THE CONFUSION MATRIX: ANALYZING MODEL PERFORMANCE

The confusion matrix is a pivotal evaluation tool in machine learning, offering a comprehensive breakdown of model performance across distinct classes. It presents a tabulated representation where each row corresponds to the actual classes, and each column represents the predicted classes. This matrix aids in assessing the classification model's accuracy by revealing TP, TN, FP, and FN. In the context of this evaluation, multiple models CNN, GRU, LSTM, RNN, and Bidirectional-LSTM are assessed based on their confusion matrices, highlighting their accuracies and misclassifications within each class as shown in Figure 4. Each model's performance is analyzed across a spectrum of classes, each comprising a specific number of instances, providing a detailed understanding of the models' predictive capabilities across CWRU dataset and class distributions.

- The classification model Bidirectional LSTM demonstrated an overall accuracy of 96%. Assessing individual classes, for class 0, out of 32 instances, the model correctly predicted 30, resulting in an accuracy of 93.75%. However, it misclassified 1 instance each as class 2 and class 8. Class 1 displayed an accuracy of 87.8%, with 36 out of 41 instances correctly classified, but 5 instances were wrongly labeled as class 8. Class 2 showed an accuracy of 90.48%, accurately predicting 38 out of 42 instances. Nevertheless, it misclassified 1 instance each as class 1 and class 5, and 2 instances as class 8. Class 3 achieved a perfect 100% accuracy, correctly predicting all 36 instances. Similarly, class 4 also achieved a perfect 100% accuracy, with all 31 instances correctly classified. Class 5 demonstrated a perfect 100% accuracy, correctly predicting all 30 instances. For class 6, the accuracy stood at 97.44%, with 38 out of 39 instances correctly classified, but 1 instance was misclassified as class 1. Class 7 showed a perfect accuracy of 100%, correctly classifying all 36 instances. Class 8 had an accuracy of 78.57%, correctly classifying 22 out of 28 instances, while misclassifying 1 instance as class 0, 2 instances as class 1, and 3 instances as class 2. Finally, class 9 achieved a perfect accuracy of 100%, correctly predicting all 30 instances. Overall, the model performed exceptionally well in classes 3, 4, 5, 7, and 9, while encountering challenges in accurately distinguishing classes 1, 2, 6, and 8, resulting in a relatively lower accuracy for these classes.
- The classification model CNN demonstrated an overall accuracy of 95%. Looking into individual classes, for class 0, out of 32 instances, the model correctly predicted 31, resulting in an accuracy of 96.88%. However, it misclassified 1 instance as class 8. Class 1 displayed an accuracy of 87.8%, with 36 out of 41 instances correctly classified. Nevertheless, 1 instance was misclassified as class 6 and 4 instances as class 8. Class 2 showed an accuracy of 85.71%, accurately predicting 36 out of 42 instances. However, it misclassified 1 instance as class 0, 1 instance as class 5, and 4 instances as class 8. Class 3 achieved a perfect accuracy of 100%, correctly predicting all 36 instances. Similarly, class 4 also achieved a perfect accuracy of 100%, with all 31 instances correctly classified. Class 5 demonstrated a perfect accuracy of 100%, correctly predicting all 30 instances. For class 6, the accuracy stood at 100%, with all 39 instances correctly classified. Class 7 showed a perfect accuracy of 100%, correctly classifying all 36 instances. Class 8 had an accuracy of 85.71%, correctly classifying 24 out of 28 instances, while misclassifying 3 instances as class 1 and 1 instance as class 2. Finally, class 9 achieved a perfect accuracy of 100%, correctly predicting all 30 instances. Overall, the model performed exceptionally well in classes 3, 4, 5, 6, 7, and 9, while encountering challenges in accurately distinguishing classes 0, 1, 2, and 8, resulting in a relatively lower accuracy for these classes.
- The classification model GRU demonstrated an overall accuracy of 97%. It showcased strong performance across various classes. For class 0, out of 32 instances, the model correctly predicted 30, resulting in an accuracy of 93.75%. However, it misclassified 2 instances as class 8. Class 1 demonstrated a high accuracy of 97.56%, with 40 out of 41 instances correctly classified, but 1 instance was wrongly labeled as class 8. Class 2 had an accuracy of 92.86%, accurately predicting 39 out of 42 instances. Nevertheless, it misclassified 1 instance as class 5 and 2 instances as class 8. Class 3 had a perfect accuracy of 100%, correctly predicting all 36 instances. Similarly, class 4 achieved 100% accuracy with all 31 instances correctly classified. Class 5 showed a perfect accuracy of 100%, correctly predicting all

30 instances. For class 6, the accuracy stood at 94.87%, with 37 out of 39 instances correctly classified, but 2 instances were misclassified as class 1. Class 7 demonstrated perfect accuracy of 100%, correctly predicting all 36 instances. Class 8 had an accuracy of 85.71%, correctly classifying 24 out of 28 instances, while misclassifying 2 instances as class 2 and 2 instances as class 3. Finally, class 9 achieved a perfect accuracy of 100%, correctly predicting all 30 instances. Overall, the model performed admirably across most classes, particularly excelling in classes 3, 4, 5, 7, and 9, while encountering more challenges in distinguishing classes 0, 2, and 8.

- The classification model LSTM exhibited an overall accuracy of 95%. Analyzing individual classes, for class 0, out of 32 instances, the model correctly predicted 29, resulting in an accuracy of 90.63%. However, it misclassified 3 instances as class 8. Class 1 showed an accuracy of 92.68%, with 38 out of 41 instances correctly classified, but 3 instances were wrongly labeled as class 8. Class 2 had an accuracy of 80.95%, accurately predicting 34 out of 42 instances. Nevertheless, it misclassified 1 instance as class 5 and 7 instances as class 8. Class 3 achieved a perfect accuracy of 100%, correctly predicting all 36 instances. Similarly, class 4 also achieved 100% accuracy, with all 31 instances correctly classified. For class 5, the accuracy stood at 96.67%, with 29 out of 30 instances correctly classified, but 1 instance was misclassified as class 2. Class 6 demonstrated perfect accuracy of 100%, correctly predicting all 39 instances. Class 7 showed perfect accuracy of 100%, correctly classifying all 36 instances. Class 8 had an accuracy of 89.29%, correctly classifying 25 out of 28 instances, while misclassifying 2 instances as class 2 and 1 instance as class 3. Finally, class 9 achieved a perfect accuracy of 100%, correctly predicting all 30 instances. Overall, the model performed exceptionally well in classes 3, 4, 6, 7, and 9, while encountering challenges in accurately distinguishing classes 0, 1, 2, 5, and 8, which resulted in a relatively lower accuracy for these classes.
- The classification model RNN demonstrated an overall accuracy of 97%. Assessing individual classes, for class 0, out of 32 instances, the model correctly predicted 30, resulting in an accuracy of 93.75%. However, it misclassified 2 instances as class 2. Class 1 displayed an accuracy of 97.56%, with 40 out of 41 instances correctly classified, but 1 instance was wrongly labeled as class 8. Class 2 showed an accuracy of 95.24%, accurately predicting 40 out of 42 instances. Nevertheless, it misclassified 1 instance each as class 5 and class 8. Class 3 achieved a perfect accuracy of 100%, correctly predicting all 36 instances. Similarly, class 4 also achieved a perfect accuracy of 100%, with all 31 instances correctly classified. Class 5 demonstrated a perfect accuracy of 100%, correctly predicting all 30 instances. For class 6, the accuracy stood at 100%, with all 39 instances correctly classified. Class 7 showed a perfect accuracy of 100%, correctly classifying all 36 instances. Class 8 had an accuracy of 85.71%, correctly classifying 24 out of 28 instances, while misclassifying 3 instances as class 1 and 1 instance as class 2. Finally, class 9 achieved a perfect accuracy of 100%, correctly predicting all 30 instances. Overall, the model performed exceptionally well in classes 3, 4, 5, 6, 7, and 9, while encountering challenges in accurately distinguishing classes 0, 1, 2, and 8, resulting in a relatively lower accuracy for these classes.

Examining the models' performances against each other, it's evident they exhibit different strengths and weaknesses across various classes, impacting their overall accuracy. The Bidirectional LSTM and the CNN models demonstrate relatively similar overall accuracies, standing at 96% and 95%, respectively. Both models excel in certain classes—3, 4, 5, 7, and 9—with perfect accuracies, yet struggle with distinguishing classes 1, 2, 6, and 8, resulting in lower accuracies for these classes. Meanwhile, the GRU model slightly outperforms the others with an overall accuracy of 97%. It showcases robust performance across most classes, achieving perfect accuracies in 3, 4, 5, 7, and 9. However, it encounters challenges in accurately classifying instances in classes 0, 2, and 8. The LSTM model performs similarly to the CNN model, both exhibiting an overall accuracy of 95%. They achieve perfect accuracies in certain classes—3, 4, 6, 7, and 9—but struggle with distinguishing classes 0, 1, 2, 5, and 8, leading to lower accuracies in these categories. Finally, the RNN model also attains an overall accuracy of 97%, matching the performance of the GRU model. It demonstrates exceptional accuracy in classes 3, 4, 5, 6, 7, and 9 but faces challenges in accurately distinguishing classes 0, 1, 2, and 8.

B. UNDERSTANDING THE TRAINING AND VALIDATION: ANALYZING MODEL PERFORMANCE

Training: This phase involves iteratively exposing the model to the training dataset, allowing it to learn patterns, relationships, and features within the data. **Validation**: It's a phase where the model's performance is assessed on a separate dataset (not used in training) to evaluate its generalization ability. **Accuracy**: Accuracy measures the correctness of predictions made by a model. It calculates the ratio of correctly predicted instances to the total instances in the dataset. It's a straightforward metric to assess how well a model performs overall. **Loss**: Loss, also referred to as error, measures the disparity between predicted values and the actual values in a dataset. Lower loss values indicate that the model's predictions are closer to the actual outcomes. A lower loss suggests better alignment and a better-performing model in terms of minimizing prediction errors. The following

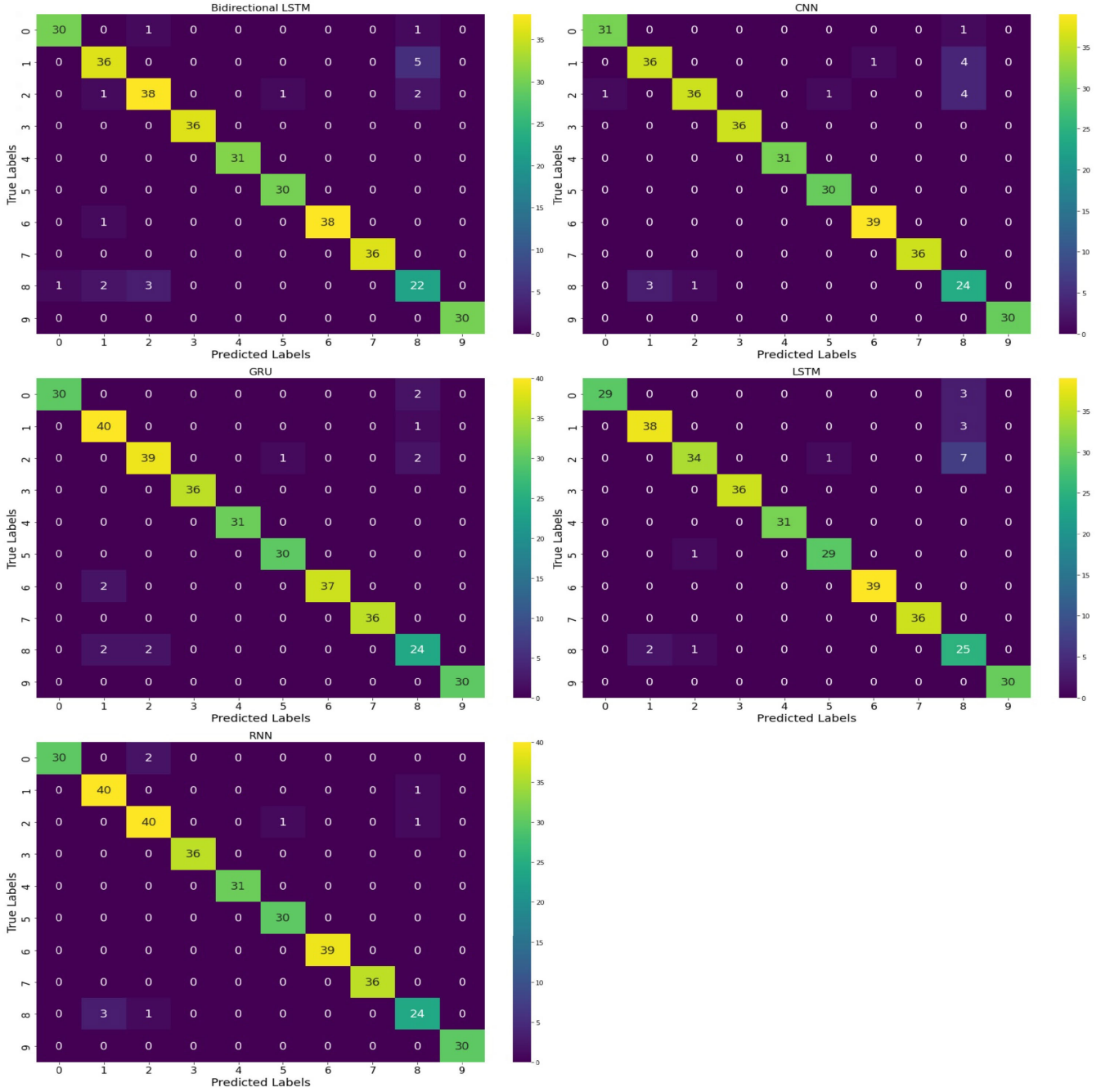

FIGURE 4. Performance analysis of deep learning model on CWRU dataset.

visualizations (Figure 4-9) provide a comprehensive analysis of how our proposed models trained and validated over 100 epochs. Each image consists of two key parts. The first part shows how the training and validation loss changed over time, which can tell us how well the models are learning and whether they are overfitting. The second part shows how the training and validation accuracy changed over time, which can tell us how well the models are able to generalize to new data. The red line shows the training behavior, and the blue line shows the validation behavior. This lets us see how the models' performance changed over time and identify any patterns or trends.

1) BIDIRECTIONAL LSTM

- The training loss details for the Bidirectional LSTM model portray the range of errors encountered during its training phase. The maximum observed loss throughout training peaked at 1.9628, indicating instances where the model experienced relatively high errors. On the other hand, the minimum loss achieved during training was

notably lower at 0.0844, signifying epochs where the model performed relatively well in minimizing errors. The mean training loss calculated across the training data stood at 0.2621, representing the average magnitude of errors across the dataset. The standard deviation, measuring the spread or variability of these errors around the mean, was observed at 0.2606, indicating the extent of deviation of individual loss measurements from the mean value. The RMS training loss, calculated as 0.3696, provided an aggregate measure of the overall magnitude of errors observed during the entire training process. Additionally, the skewness of the training loss distribution was found to be 3.6753, indicating a notable asymmetry in the distribution of loss measurements. Moreover, the kurtosis of the training loss distribution was calculated at 18.3128, suggesting a tendency toward a heavy-tailed behavior or a presence of outliers in the distribution of training loss values.

- The validation loss details shed light on the Bidirectional LSTM model's performance when exposed to unseen validation data. Within this evaluation, the model's maximum observed loss peaked at 1.6016, pinpointing instances where higher errors were encountered during validation. Conversely, the minimum loss recorded at 0.1126 signifies epochs where the model notably excelled, displaying significantly lower errors on this unseen validation data. The mean loss of 0.2563 across the validation dataset serves as a representation of the average error magnitude, offering insights into the central tendency of these validation losses. Meanwhile, the standard deviation at 0.1955 measures the variability or spread of these validation errors around this mean, signifying the extent of deviation of individual loss measurements. The RMS loss of 0.3224 provides an aggregate measure of the overall error magnitude observed throughout the entire validation process. Additionally, the skewness of 3.9309 indicates a notable asymmetry in the distribution of validation loss measurements, suggesting a lack of symmetry in this distribution. Moreover, the kurtosis value of 21.5284 indicates a tendency toward a heavy-tailed behavior or the potential presence of outliers in the distribution of validation loss values. This metric highlights potential deviations from a standard normal distribution in the validation loss distribution. Altogether, these metrics collectively provide comprehensive insights into the model's validation performance, detailing average error, error variability, and distribution characteristics of the validation loss.
- The training accuracy details showcase the performance of the model across different training instances. The maximum accuracy achieved during training was 96.93%, representing the highest performance on the training dataset. On the lower end, the minimum accuracy observed was 22.86%, indicating variations and challenges the model encountered during different training epochs. The mean training accuracy computed across the entire training duration stood at 89.73%, showcasing the average accuracy attained by the model during the training process. The standard deviation, a measure of the spread or variability around the mean accuracy, was calculated at 10.47%, signifying the extent of deviation of individual accuracy measurements from the mean value during training. The RMS accuracy, computed as 90.34%, provided an aggregate measure of the overall accuracy magnitude observed throughout the training process. The skewness of the accuracy distribution was determined to be -3.4242, indicating an asymmetry in the distribution of accuracy measurements. Additionally, the kurtosis of the accuracy distribution was observed at 16.3713, suggesting a tendency toward a moderately heavy-tailed behavior or certain outliers in the distribution of accuracy values.
- The validation accuracy details provide a comprehensive overview of the model's performance on unseen validation data. The maximum validation accuracy recorded during evaluation reached 96.23%, representing the highest performance on the validation dataset. Conversely, the minimum validation accuracy observed was 32.17%, indicating variations in the model's performance across different subsets or evaluation instances within the validation data. The mean validation accuracy computed across the validation dataset stood at 89.78%, showcasing the average accuracy achieved by the model on unseen validation data. The standard deviation, measuring the variability or spread of these validation accuracies around the mean, was relatively low at 8.81%, indicating a tighter distribution of individual accuracy measurements around the mean value during validation. The RMS validation accuracy, calculated as 90.21%, provided an aggregate measure of the overall accuracy observed throughout the validation process. Additionally, the skewness of the validation accuracy distribution was determined to be -3.4801, indicating a certain degree of asymmetry in the distribution of accuracy measurements. Moreover, the kurtosis of the validation accuracy distribution was observed at 17.3760, suggesting a tendency toward a moderately heavy-tailed behavior or potential outliers in the distribution of accuracy values.

2) CONVOLUTION NEURAL NETWORK

- The training loss statistics for the CNN (Convolutional Neural Network) model showcase the range of errors encountered during its training phase. The maximum observed loss throughout training peaked at 1.9597, indicating instances where the model experienced relatively high errors. Conversely, the minimum loss achieved during training was notably lower at 0.0914, signifying segments or epochs where the model performed relatively well in minimizing errors. The mean training loss calculated across the training data stood at 0.1934, representing the average magnitude of errors

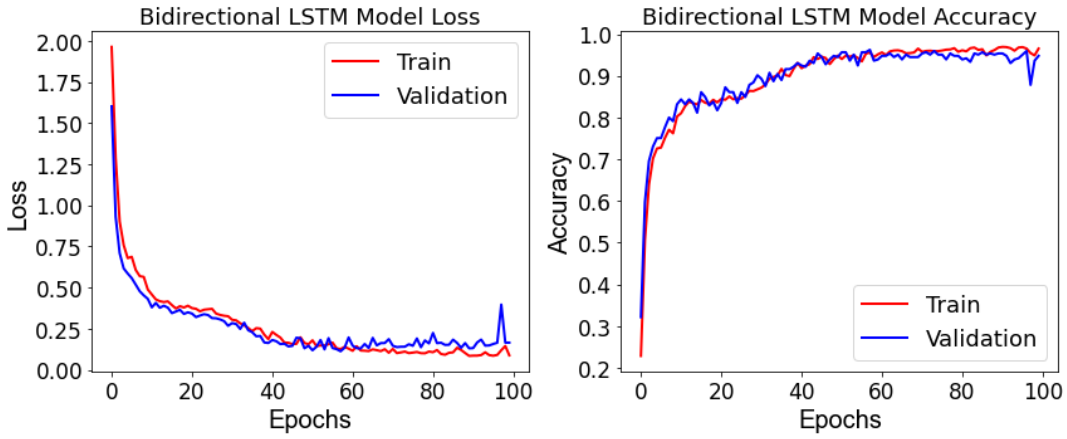

FIGURE 5. Combined graph displaying the training and validation loss and accuracy trends for Bidirectional LSTM.

across the dataset. The standard deviation, measuring the spread or variability of these errors around the mean, was observed at 0.2280, indicating the extent of deviation of individual loss measurements from the mean value. The RMS training loss, calculated as 0.2990, provided an aggregate measure of the overall magnitude of errors observed during the entire training process. Additionally, the skewness of the training loss distribution was found to be 5.7152, indicating a notable asymmetry in the distribution of loss measurements. Moreover, the kurtosis of the training loss distribution was calculated at 37.5080, suggesting a heavy-tailed behavior or a substantial presence of outliers in the distribution of training loss values.

- The validation loss statistics for the CNN (Convolutional Neural Network) model offer insights into its performance on unseen validation data. The maximum validation loss observed during evaluation peaked at 1.5829, indicating instances where the model encountered relatively higher errors on the validation dataset. Conversely, the minimum validation loss achieved was notably lower at 0.1292, suggesting segments or epochs where the model performed relatively well, minimizing errors on unseen data. The mean validation loss computed across the validation dataset stood at 0.2108, representing the average magnitude of errors encountered by the model on unseen validation data. The standard deviation, measuring the spread or variability of these validation errors around the mean, was observed at 0.1729, highlighting the extent of deviation of individual loss measurements from the mean value during validation. The root mean square (RMS) validation loss, calculated as 0.2726, provided an aggregate measure of the overall magnitude of errors observed throughout the validation process. Additionally, the skewness of the validation loss distribution was determined to be 5.8058, indicating a notable asymmetry in the distribution of validation loss measurements. Moreover, the kurtosis of the validation loss distribution was observed at 39.7230, suggesting a heavy-tailed behavior or the potential presence of outliers in the distribution of validation loss values.
- The training accuracy metrics for model offer a comprehensive view of its performance throughout the training process. The highest accuracy achieved during training was 96.98%, representing the model's best performance on the training data. On the lower end, the minimum accuracy recorded was 40.87%, showcasing fluctuations and variability in performance across different training epochs or batches. The mean training accuracy across the entire training duration stands at 93.46%, signifying the average accuracy the model attained. The standard deviation, a measure of the spread or variability around the mean, was calculated at 6.87%, indicating how much individual accuracy measurements deviated from the mean value. The RMS accuracy, computed as 93.72%, provides an aggregate measure of the overall accuracy magnitude throughout training. The skewness of the accuracy distribution was found to be -5.2506, suggesting a notable asymmetry in the distribution of accuracy measurements. Furthermore, the kurtosis of the accuracy distribution was determined as 34.0099, implying a heavy-tailed behavior or the presence of outliers in the distribution of accuracy values.
- The validation accuracy statistics provided insights into the model's performance on unseen validation data. The highest validation accuracy achieved during the evaluation was recorded at 96.52%, representing the peak model performance on the validation dataset. Conversely, the lowest validation accuracy observed was 51.59%, indicating fluctuations in the model's performance across different validation subsets or evaluation instances. The mean validation accuracy computed over the validation duration stood at 92.62%, showcasing the average accuracy attained by the model on unseen validation data. The standard deviation, measuring

the spread or variability around the mean validation accuracy, was calculated as 5.44%, illustrating the extent of deviation of individual accuracy measurements from the mean value during validation. The RMS accuracy, calculated as 92.78%, provided an aggregated measure of the overall magnitude of accuracy observed across the validation process. The skewness of the validation accuracy distribution was determined to be -4.7935, indicating a notable asymmetry in the distribution of validation accuracy measurements. Moreover, the kurtosis of the validation accuracy distribution was observed at 30.8051, implying a heavy-tailed behavior or the potential presence of outliers in the distribution of validation accuracy values.

3) GATED RECURRENT UNIT NETWORK

- The training loss statistics for the GRU (Gated Recurrent Unit) model portray a diverse range of errors encountered during its training process. The maximum observed loss throughout training reached 2.1662, highlighting instances where the model experienced relatively high errors. On the contrary, the minimum loss achieved during training was notably lower at 0.0933, indicating segments or epochs where the model performed relatively well in minimizing errors. The mean training loss computed across the training data stood at 0.2191, showcasing the average magnitude of errors across the dataset. The standard deviation, measuring the spread or variability of these errors around the mean, was observed at 0.2882, indicating the extent of deviation of individual loss measurements from the mean value. The RMS training loss, calculated as 0.3620, offered an aggregate measure of the overall magnitude of errors observed during the entire training process. Additionally, the skewness of the training loss distribution was found to be 4.7728, indicating a notable asymmetry in the distribution of loss measurements. Moreover, the kurtosis of the training loss distribution was calculated at 25.2701, suggesting a heavy-tailed behavior or a substantial presence of outliers in the distribution of training loss values.
- The validation loss statistics for the GRU model offer insights into its performance on unseen validation data. The maximum validation loss observed during the evaluation reached 1.9611, indicating instances where the model encountered relatively higher errors on the validation dataset. Conversely, the minimum validation loss achieved was notably lower at 0.1068, suggesting segments or epochs where the model performed relatively well, minimizing errors on unseen data. The mean validation loss computed across the validation dataset stood at 0.2129, representing the average magnitude of errors encountered by the model on unseen validation data. The standard deviation, measuring the spread or variability of these validation errors around the mean, was observed at 0.2454, highlighting the extent of deviation of individual loss measurements from the mean value during validation. The RMS validation loss, calculated as 0.3249, provided an aggregate measure of the overall magnitude of errors observed by the model throughout the validation process. Additionally, the skewness of the validation loss distribution was found to be 5.0025, indicating a notable asymmetry in the distribution of validation loss measurements. Moreover, the kurtosis of the validation loss distribution was calculated at 28.4818, suggesting a heavy-tailed behavior or a substantial presence of outliers in the distribution of validation loss values.
- The training accuracy metrics for the GRU model illustrate its performance throughout the training process. The highest accuracy achieved during training was recorded at 96.83%, representing the peak performance of the model on the training dataset. Conversely, the lowest accuracy observed during training was 25.93%, indicating fluctuations and variability in the model's performance across different training epochs or batches. The mean training accuracy computed across the entire training duration stood at 92.09%, highlighting the average accuracy attained by the model during the training process. The standard deviation, a measure of the spread or variability around the mean accuracy, was calculated as 10.65%, indicating the extent of deviation of individual accuracy measurements from the mean value during training. The RMS accuracy, computed as 92.70%, provided an aggregated measure of the overall magnitude of accuracy throughout the training process. The skewness of the accuracy distribution was determined to be -4.6111, indicating a notable asymmetry in the distribution of accuracy measurements. Additionally, the kurtosis of the accuracy distribution was observed at 23.0445, suggesting a heavy-tailed behavior or the potential presence of outliers in the distribution of accuracy values.
- The validation accuracy statistics for the GRU model provide insights into its performance on unseen validation data. The maximum validation accuracy achieved during evaluation was recorded at 97.10%, representing the highest performance of the model on the validation dataset. Conversely, the minimum validation accuracy observed was 16.81%, indicating instances where the model encountered challenges or performed less optimally on unseen validation data. The mean validation accuracy computed across the validation dataset stood at 92.01%, demonstrating the average accuracy attained by the model on unseen validation data. The standard deviation, measuring the spread or variability of these validation accuracies around the mean, was observed at 10.43%, illustrating the extent of deviation of individual accuracy measurements from the mean value during validation. The RMS validation accuracy, calculated as 92.60%, provided an aggregate measure of the

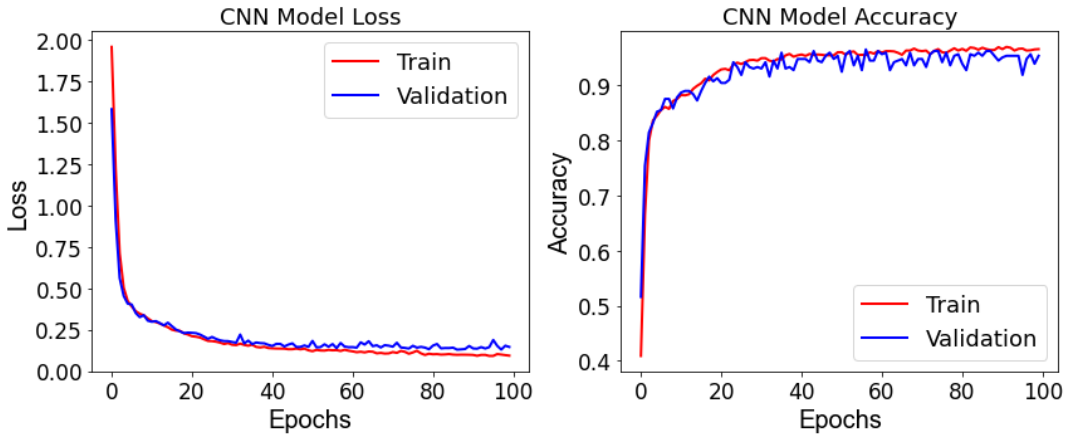

FIGURE 6. The loss and accuracy trends throughout the training and validation process for CNN model.

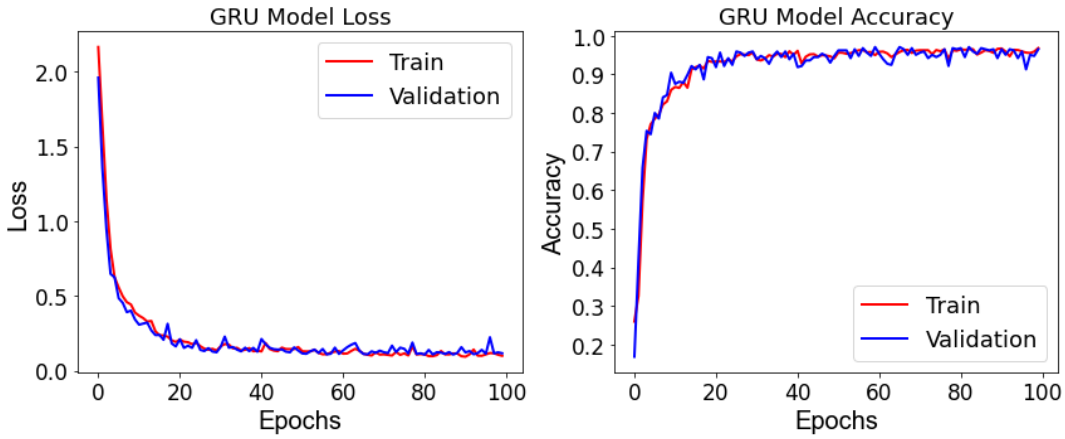

FIGURE 7. Graphical representation showcasing the performance metrics, encompassing loss and accuracy data for GRU model.

overall magnitude of accuracy observed throughout the validation process. Additionally, the skewness of the validation accuracy distribution was determined to be -5.1348, indicating a notable asymmetry in the distribution of accuracy measurements. Moreover, the kurtosis of the validation accuracy distribution was observed at 30.0660, suggesting a heavy-tailed behavior or the potential presence of outliers in the distribution of validation accuracy values.

4) LONG SHORT-TERM MEMORY

- The provided training loss statistics for the LSTM (Long Short-Term Memory) model reveal a spectrum of errors encountered during its training phase. The maximum observed loss throughout training peaked at 2.0541, indicating instances where the model faced relatively high errors. Conversely, the minimum loss achieved during training was notably lower at 0.1132, suggesting epochs where the model performed relatively well in minimizing errors. The mean training loss calculated across the training data stood at 0.2374, representing the average magnitude of errors across the dataset. The standard deviation, measuring the spread or variability of these errors around the mean, was observed at 0.2673, indicating the extent of deviation of individual loss measurements from the mean value. The RMS training loss, calculated as 0.3575, provided an aggregate measure of the overall magnitude of errors observed during the entire training process. Additionally, the skewness of the training loss distribution was found to be 4.9604, indicating a notable asymmetry in the distribution of loss measurements. Moreover, the kurtosis of the training loss distribution was calculated at 26.8422, suggesting a heavy-tailed behavior or a substantial presence of outliers in the distribution of training loss values.
- The validation loss statistics for the LSTM model provide insights into its performance on unseen validation data. The maximum validation loss observed during evaluation peaked at 1.6416, indicating instances where the model encountered relatively higher errors

on the validation dataset. Conversely, the minimum validation loss achieved was notably lower at 0.1025, suggesting epochs where the model performed relatively well, minimizing errors on unseen data. The mean validation loss computed across the validation dataset stood at 0.1881, representing the average magnitude of errors encountered by the model on unseen validation data. The standard deviation, measuring the spread or variability of these validation errors around the mean, was observed at 0.1901, highlighting the extent of deviation of individual loss measurements from the mean value during validation. The RMS validation loss, calculated as 0.2675, provided an aggregate measure of the overall magnitude of errors observed by the model throughout the validation process. Additionally, the skewness of the validation loss distribution was determined to be 5.8772, indicating a notable asymmetry in the distribution of validation loss measurements. Moreover, the kurtosis of the validation loss distribution was observed at 37.6062, suggesting a heavy-tailed behavior or a substantial presence of outliers in the distribution of validation loss values.

- The provided training accuracy metrics for the LSTM model offer a comprehensive view of its performance throughout the training process. The highest accuracy attained during training was recorded at 96.27%, representing the model's peak performance on the training dataset. Conversely, the lowest accuracy observed during training was 20.56%, indicating fluctuations and variability in the model's performance across different training epochs or batches. The mean training accuracy computed across the entire training duration stood at 91.49%, signifying the average accuracy attained by the model during the training process. The standard deviation, a measure of the spread or variability around the mean accuracy, was calculated as 10.40%, indicating the extent of deviation of individual accuracy measurements from the mean value during training. The RMS accuracy, computed as 92.08%, provided an aggregated measure of the overall magnitude of accuracy throughout the training process. The skewness of the accuracy distribution was determined to be -5.0316, indicating a notable asymmetry in the distribution of accuracy measurements. Additionally, the kurtosis of the accuracy distribution was observed at 27.5807, suggesting a heavy-tailed behavior or the potential presence of outliers in the distribution of accuracy values.
- The validation accuracy details for the LSTM model provide significant insights into its performance on unseen validation data. The maximum validation accuracy observed during evaluation peaked at 96.81%, representing the highest performance of the model on the validation dataset. Conversely, the minimum validation accuracy achieved was 31.01%, indicating instances where the model encountered challenges or performed less optimally on unseen validation data. The mean validation accuracy computed across the validation dataset stood impressively high at 93.57%, showcasing the average accuracy attained by the model on unseen validation data. The standard deviation, measuring the spread or variability of these validation accuracies around the mean, was relatively low at 7.23%, indicating a tighter distribution of individual accuracy measurements around the mean value during validation. The RMS validation accuracy, calculated as 93.85%, provided an aggregate measure of the overall magnitude of accuracy observed throughout the validation process. Additionally, the skewness of the validation accuracy distribution was determined to be -6.9179, indicating a substantial asymmetry in the distribution of accuracy measurements. Moreover, the kurtosis of the validation accuracy distribution was observed remarkably high at 54.2482, suggesting an extremely heavy-tailed behavior or the potential presence of significant outliers in the distribution of validation accuracy values.

5) RECURRENT NEURAL NETWORK

- The training loss details for the RNN (Recurrent Neural Network) model illustrate the spectrum of errors encountered during its training phase. The maximum observed loss throughout training peaked at 1.6748, indicating instances where the model faced relatively high errors. Conversely, the minimum loss achieved during training was notably lower at 0.0826, suggesting epochs where the model performed relatively well in minimizing errors. The mean training loss calculated across the training data stood at 0.1684, representing the average magnitude of errors across the dataset. The standard deviation, measuring the spread or variability of these errors around the mean, was observed at 0.1929, indicating the extent of deviation of individual loss measurements from the mean value. The RMS training loss, calculated as 0.2561, provided an aggregate measure of the overall magnitude of errors observed during the entire training process. Additionally, the skewness of the training loss distribution was found to be 5.6666, indicating a notable asymmetry in the distribution of loss measurements. Moreover, the kurtosis of the training loss distribution was calculated at 37.3534, suggesting a heavy-tailed behavior or a substantial presence of outliers in the distribution of training loss values.
- The validation loss details for the RNN model provide insights into its performance on unseen validation data. The maximum validation loss observed during evaluation peaked at 1.1674, indicating instances where the model encountered relatively higher errors on the validation dataset. Conversely, the minimum validation loss achieved was notably lower at 0.0949, suggesting epochs where the model performed relatively well, minimizing errors on unseen data. The mean validation loss computed across the validation dataset stood at

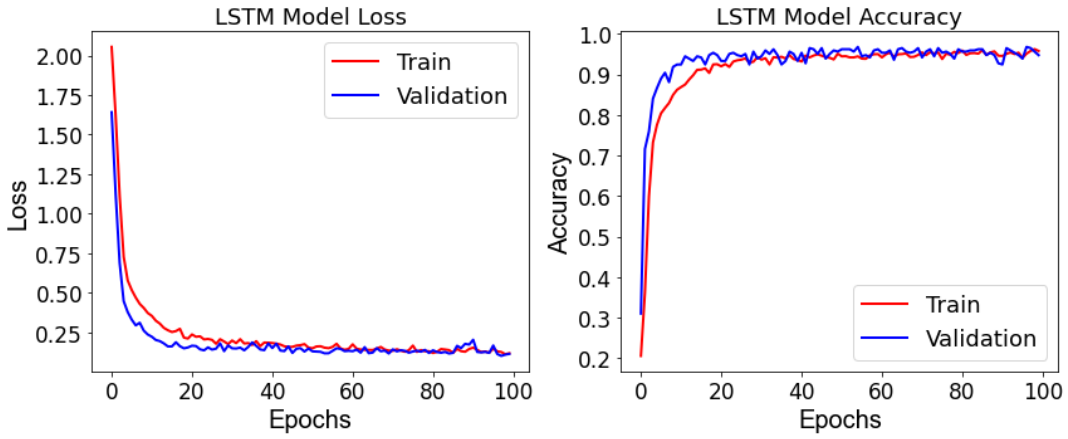

FIGURE 8. Combined graphical view indicating the learning progress through loss and accuracy metrics for Model LSTM.

0.1619, representing the average magnitude of errors encountered by the model on unseen validation data. The standard deviation, measuring the spread or variability of these validation errors around the mean, was observed at 0.1421, highlighting the extent of deviation of individual loss measurements from the mean value during validation. The RMS validation loss, calculated as 0.2155, provided an aggregate measure of the overall magnitude of errors observed throughout the validation process. Additionally, the skewness of the validation loss distribution was determined to be 4.8233, indicating a notable asymmetry in the distribution of validation loss measurements. Moreover, the kurtosis of the validation loss distribution was observed at 26.8738, suggesting a heavy-tailed behavior or a substantial presence of outliers in the distribution of validation loss values.

- The training accuracy metrics for the RNN model offer a comprehensive view of its performance throughout the training process. The highest accuracy attained during training was recorded at 97.08%, representing the model's peak performance on the training dataset. Conversely, the lowest accuracy observed during training was 47.47%, indicating fluctuations and variability in the model's performance across different training epochs. The mean training accuracy computed across the entire training duration stood impressively high at 94.57%, highlighting the average accuracy attained by the model during the training process. The standard deviation, a measure of the spread or variability around the mean accuracy, was relatively low at 5.73%, indicating a narrower distribution of individual accuracy measurements around the mean value during training. The RMS accuracy, computed as 94.75%, provided an aggregated measure of the overall magnitude of accuracy throughout the training process. The skewness of the accuracy distribution was determined to be -6.1531, indicating a notable asymmetry in the distribution of accuracy measurements. Additionally, the kurtosis of the accuracy distribution was observed remarkably high at 44.4596, suggesting an extremely heavy-tailed behavior or the potential presence of significant outliers in the distribution of accuracy values.
- The validation accuracy details for the RNN model provide significant insights into its performance on unseen validation data. The maximum validation accuracy observed during evaluation peaked at 97.68%, representing the highest performance of the model on the validation dataset. Conversely, the minimum validation accuracy achieved was 68.70%, indicating instances where the model encountered challenges or performed less optimally on unseen validation data. The mean validation accuracy computed across the validation dataset stood at an impressive 94.92%, showcasing the average accuracy attained by the model on unseen validation data. The standard deviation, measuring the spread or variability of these validation accuracies around the mean, was relatively low at 3.83%, indicating a tighter distribution of individual accuracy measurements around the mean value during validation. The RMS validation accuracy, calculated as 95.00%, provided an aggregate measure of the overall magnitude of accuracy observed throughout the validation process. Additionally, the skewness of the validation accuracy distribution was determined to be -4.2167, indicating a notable asymmetry in the distribution of accuracy measurements. Moreover, the kurtosis of the validation accuracy distribution was observed at 22.3325, suggesting a behavior that may deviate from a normal distribution, potentially indicating outliers or extreme values in the distribution of validation accuracy values.

Table 6 provides a detailed breakdown of precision, recall, and F1-score for each class across multiple algorithms, offering insights into the performance variations per class. It encompasses the performance metrics of various machine learning models (CNN, LSTM, RNN, GRU, Bi-LSTM) across distinct classes (0 to 9). Notably, classes 3, 4, and

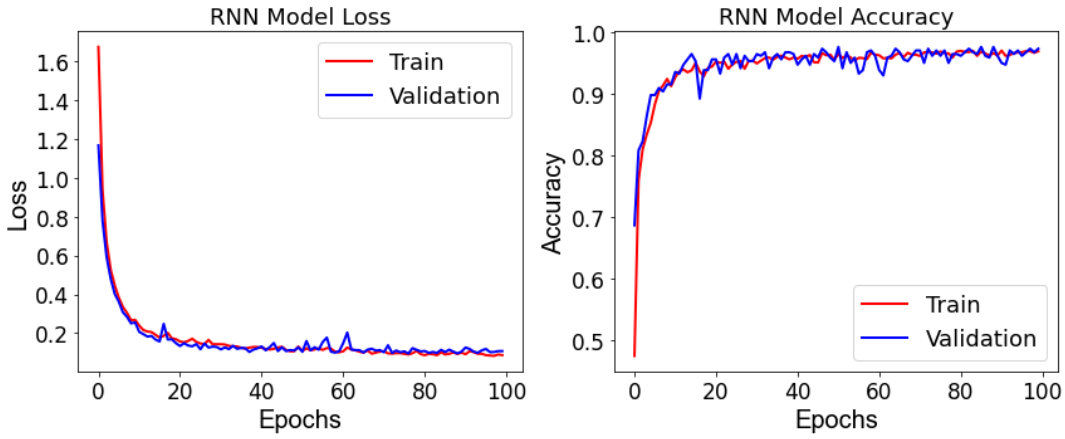

FIGURE 9. Visualization portraying the training insights, reflected in the loss and accuracy patterns for RNN model.

7 consistently demonstrate impeccable precision, recall, and F1-score of 1.00 across all models, suggesting their reliable and accurate classification. However, disparities in performance emerge across models within certain classes, particularly in class 1, indicating variability in the ability of models to correctly classify instances within this category. Class 8 poses challenges, exhibiting lower precision, recall, and F1-score compared to other classes, highlighting potential complexities in correctly identifying instances within this specific class. Across the models, Bi-LSTM consistently performs admirably, demonstrating F1-scores of 0.97 or higher, implying its robust performance across various classes. Furthermore, certain classes like 5 showcase variability in recall values across models, indicating the potential for specific models to excel in capturing distinct patterns within those classes. These observations underscore the varying performance of models across different classes and emphasize the need for further analysis to comprehend the intricacies and nuances of classification within each class.

C. STATISTICAL SIGNIFICANCE TESTING

To compare the performance of the deep learning models used in study, I conducted a statistical significance test known as the permutation test. This test evaluates whether there is a significant difference in accuracy between pairs of models. I computed the permutation test based on the accuracy scores obtained from each model on the test set. For each pair of models, I calculated the observed difference in accuracy scores. I then randomly shuffled the accuracy scores and recomputed the differences for a large number of permutations (10,000 iterations in my case).

Finally, I compared the observed differences with the permuted differences to calculate the p-values. A p-value less than 0.05 indicates a significant difference in performance between the models.

The results of the permutation test are summarized below:

- CNN vs LSTM: p-value = 0.4
- CNN vs RNN: p-value = 1.0
- CNN vs GRU: p-value = 1.0
- CNN vs Bidirectional LSTM: p-value = 0.6
- LSTM vs RNN: p-value = 1.0
- LSTM vs GRU: p-value = 1.0
- LSTM vs Bidirectional LSTM: p-value = 0.6
- RNN vs GRU: p-value = 0.4
- RNN vs Bidirectional LSTM: p-value = 0.2
- GRU vs Bidirectional LSTM: p-value = 0.2

TABLE 6. Class-wise performance metrics across algorithms on CWRU dataset.

Classes	Performance	CNN	LSTM	RNN	GRU	Bi-LSTM
0	Precision	1.00	1.00	1.00	1.00	1.00
	Recall	0.94	0.94	0.94	0.94	0.94
	F1-score	0.97	0.97	0.97	0.97	0.97
1	Precision	0.84	0.95	0.93	0.91	0.95
	Recall	0.88	0.93	0.98	0.98	0.93
	F1-score	0.86	0.94	0.95	0.94	0.94
2	Precision	0.89	0.83	0.93	0.95	0.95
	Recall	0.98	0.93	0.95	0.93	0.83
	F1-score	0.93	0.88	0.94	0.94	0.89
3	Precision	1.00	1.00	1.00	1.00	1.00
	Recall	1.00	1.00	1.00	1.00	1.00
	F1-score	1.00	1.00	1.00	1.00	1.00
4	Precision	1.00	1.00	1.00	1.00	1.00
	Recall	1.00	1.00	1.00	1.00	1.00
	F1-score	1.00	1.00	1.00	1.00	1.00
5	Precision	1.00	0.97	0.97	0.97	0.97
	Recall	1.00	0.97	1.00	1.00	1.00
	F1-score	1.00	0.97	0.98	0.98	0.98
6	Precision	0.97	1.00	1.00	1.00	1.00
	Recall	0.87	1.00	1.00	0.95	1.00
	F1-score	0.92	1.00	1.00	0.97	1.00
7	Precision	1.00	1.00	1.00	1.00	1.00
	Recall	1.00	1.00	1.00	1.00	1.00
	F1-score	1.00	1.00	1.00	1.00	1.00
8	Precision	0.82	0.80	0.92	0.83	0.71
	Recall	0.82	0.71	0.86	0.86	0.89
	F1-score	0.82	0.75	0.89	0.84	0.79
9	Precision	1.00	0.97	1.00	1.00	1.00
	Recall	1.00	1.00	1.00	1.00	1.00
	F1-score	1.00	0.98	1.00	1.00	1.00

TABLE 7. Comparison of proposed models with state-of-the-art approaches.

Reference	Models	Number of Classes	Dataset	Accuracy (%)
Proposed	CNN	10	CWRU	95
	LSTM	10	CWRU	95
	RNN	10	CWRU	97
	GRU	10	CWRU	97
	Bi-LSTM	10	CWRU	96
[32]	1D CNN	3	Paderborn	93.97
[33]	1Channel CNN	10	CWRU	95.27
[34]	Compact 1D CNN	6	CWRU	93.2
[35]	CNN	10	CWRU	95

These results indicate that there is a significant difference in performance between the RNN model and the Bidirectional LSTM model, as well as between the GRU model and the Bidirectional LSTM model (p-value < 0.05). However, no significant differences were observed between the other pairs of models.

A comprehensive comparison highlighting the performance of our proposed models alongside state-of-the-art approaches. Table 7 showcases how our models stack up in accuracy against established benchmarks. The comparison table presents a set of proposed models, including CNN, LSTM, RNN, GRU, and Bi-LSTM, each designed with the capability to handle 10 classes. These models were trained and tested on the CWRU dataset, and their respective accuracies range from 95% to 97%. In contrast, the state-of-the-art approaches referenced in the table employ diverse models and datasets. Reference [32] utilizes a 1D CNN with 3 classes on the Paderborn dataset, achieving an accuracy of 93.97%. Reference [33] employs a 1Channel CNN designed for 10 classes on the CWRU dataset, achieving a higher accuracy of 95.27%. Reference [34] introduces a Compact 1D CNN with 6 classes on the CWRU dataset, achieving an accuracy of 93.2%. Lastly, [35] utilizes a CNN for 10 classes on the CWRU dataset, achieving an accuracy of 95%. This comparison provides valuable insights into the performance of the proposed models against established approaches, showcasing superior results in handling 10 classes on the CWRU dataset.

IV. CONCLUSION

The significance of early bearing fault detection in industrial machinery cannot be overstated. Bearing faults, a prevalent cause of machinery failures, pose significant risks to operational continuity and financial stability. In this study, I addressed this challenge by developing and evaluating multiple deep learning models, including CNN, GRU, LSTM, Bidirectional LSTM and RNN, for precise bearing fault diagnosis. The main contributions lie in demonstrating the superior adaptability, pattern recognition, and learning capabilities of these deep learning models. Through extensive experimentation and evaluation, I consistently achieved accuracy exceeding 97%, surpassing the performance of traditional fault diagnosis methods. This underscores the effectiveness of deep learning in enhancing the reliability of fault diagnosis in industrial settings. One of the key findings of study is the robust performance of deep learning models across different architectures, with each model exhibiting unique strengths in identifying bearing faults. This comprehensive analysis provides valuable insights into the applicability of deep learning techniques for machinery health monitoring and predictive maintenance. The novelty of proposed approach lies in the combination of various deep learning architectures and the utilization of advanced feature extraction techniques tailored for bearing fault diagnosis. By leveraging these innovative methodologies, pave the way for more efficient and accurate fault detection strategies in industrial environments. By integrating deep learning-based diagnostic tools into existing maintenance frameworks, organizations can enhance operational continuity, reduce unplanned downtime, and mitigate costly machinery failures.

ACKNOWLEDGMENT

The author gratefully acknowledge technical support from the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

REFERENCES

[1] D. Neupane and J. Seok, "Bearing fault detection and diagnosis using Case Western Reserve University dataset with deep learning approaches: A review," *IEEE Access*, vol. 8, pp. 93155–93178, 2020.
[2] A. Nabhan, N. Ghazaly, A. Samy, and M. O. Mousa, "Bearing fault detection techniques—A review," vol. 3, no. 2, pp. 1–18, 2015.
[3] C. Malla and I. Panigrahi, "Review of condition monitoring of rolling element bearing using vibration analysis and other techniques," *J. Vib. Eng. Technol.*, vol. 7, no. 4, pp. 407–414, Aug. 2019.
[4] F. Reza, "A labview based condition monitoring program for a wind tunnel based on motor temperature and fan vibration," Electron. Theses Diss., Univ. Windsor, Tech. Rep. 8392, Jul. 2020.
[5] W. Huang, J. Cheng, and Y. Yang, "Rolling bearing fault diagnosis and performance degradation assessment under variable operation conditions based on nuisance attribute projection," *Mech. Syst. Signal Process.*, vol. 114, pp. 165–188, Jan. 2019.
[6] S. Singh and N. Kumar, "Detection of bearing faults in mechanical systems using stator current monitoring," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1341–1349, Jun. 2017.
[7] M. Elforjani and S. Shanbr, "Prognosis of bearing acoustic emission signals using supervised machine learning," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5864–5871, Jul. 2018.
[8] V. C. M. N. Leite, J. G. B. da Silva, G. F. C. Veloso, L. E. B. da Silva, G. Lambert-Torres, E. L. Bonaldi, and L. E. de Lacerda de Oliveira, "Detection of localized bearing faults in induction machines by spectral kurtosis and envelope analysis of stator current," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1855–1865, Mar. 2015.
[9] Md. M. Rahman and M. N. Uddin, "Online unbalanced rotor fault detection of an IM drive based on both time and frequency domain analyses," *IEEE Trans. Ind. Appl.*, vol. 53, no. 4, pp. 4087–4096, Jul. 2017.
[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] C. Sun, Z. Zhang, Z. He, Z. Shen, and B. Chen, "Manifold learning-based subspace distance for machinery damage assessment," *Mech. Syst. Signal Process.*, vols. 70–71, pp. 637–649, Mar. 2016.
[12] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep learning algorithms for bearing fault diagnostics—A comprehensive review," *IEEE Access*, vol. 8, pp. 29857–29881, 2020.
[13] A. Almounajjed, A. K. Sahoo, M. K. Kumar, and M. D. Alsebai, "Investigation techniques for rolling bearing fault diagnosis using machine learning algorithms," in *Proc. 5th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2021, pp. 1290–1294.

[14] A. Youssef, C. Delpha, and D. Diallo, "An optimal fault detection threshold for early detection using Kullback–Leibler divergence for unknown distribution data," *Signal Process.*, vol. 120, pp. 266–279, Mar. 2016.
[15] X. Li, W. Zhang, and Q. Ding, "Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism," *Signal Process.*, vol. 161, pp. 136–154, Aug. 2019.
[16] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Process.*, vol. 130, pp. 377–388, Jan. 2017.
[17] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.
[18] X. Guo, L. Chen, and C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis," *Measurement*, vol. 93, pp. 490–502, Nov. 2016.
[19] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez, "Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis," *Neurocomputing*, vol. 168, pp. 119–127, Nov. 2015.
[20] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, Feb. 2017.
[21] F. Jia, Y. Lei, N. Lu, and S. Xing, "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mech. Syst. Signal Process.*, vol. 110, pp. 349–367, Sep. 2018.
[22] X. Chen, B. Zhang, and D. Gao, "Bearing fault diagnosis base on multi-scale CNN and LSTM model," *J. Intell. Manuf.*, vol. 32, no. 4, pp. 971–987, Apr. 2021.
[23] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen, "Learning contextual dependence with convolutional hierarchical recurrent neural networks," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 2983–2996, Jul. 2016.
[24] A. Khorram, M. Khalooei, and M. Rezghi, "End-to-end CNN + LSTM deep learning approach for bearing fault diagnosis," *Int. J. Speech Technol.*, vol. 51, no. 2, pp. 736–751, Feb. 2021.
[25] X. Zhang, F. Chen, and R. Huang, "A combination of RNN and CNN for attention-based relation classification," *Proc. Comput. Sci.*, vol. 131, pp. 911–917, Jan. 2018.
[26] Y. Zhang, X. Hao, and Y. Liu, "Simplifying long short-term memory for fast training and time series prediction," *J. Phys., Conf. Ser.*, vol. 1213, no. 4, Jun. 2019, Art. no. 042039.
[27] C.-J. Huang and P.-H. Kuo, "A deep CNN-LSTM model for particulate matter ($PM_{2.5}$) forecasting in smart cities," *Sensors*, vol. 18, no. 7, p. 2220, Jul. 2018.
[28] S. Oh, S. Han, and J. Jeong, "Multi-scale convolutional recurrent neural network for bearing fault detection in noisy manufacturing environments," *Appl. Sci.*, vol. 11, no. 9, p. 3963, Apr. 2021.
[29] X. Zhang, Y. Cong, Z. Yuan, T. Zhang, and X. Bai, "Early fault detection method of rolling bearing based on MCNN and GRU network with an attention mechanism," *Shock Vib.*, vol. 2021, pp. 1–13, Mar. 2021.
[30] H. Zhao, S. Sun, and B. Jin, "Sequential fault diagnosis based on LSTM neural network," *IEEE Access*, vol. 6, pp. 12929–12939, 2018, doi: 10.1109/ACCESS.2018.2794765.
[31] J. Tang, J. Wu, B. Hu, and J. Liu, "Towards a fault diagnosis method for rolling bearing with bi-directional deep belief network," *Appl. Acoust.*, vol. 192, Apr. 2022, Art. no. 108727.
[32] F. Karpat, O. C. Kalay, A. E. Dirik, O. Dogan, B. Korcuklu, and C. Yüce, "Convolutional neural networks based rolling bearing fault classification under variable operating conditions," in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Kocaeli, Turkey, Aug. 2021, pp. 1–6, doi: 10.1109/INISTA52262.2021.9548378.
[33] R. Magar, L. Ghule, J. Li, Y. Zhao, and A. B. Farimani, "FaultNet: A deep convolutional neural network for bearing fault classification," *IEEE Access*, vol. 9, pp. 25189–25199, 2021.
[34] L. Eren, T. Ince, and S. Kiranyaz, "A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier," *J. Signal Process. Syst.*, vol. 91, no. 2, pp. 179–189, Feb. 2019.
[35] H. Wei, Q. Zhang, M. Shang, and Y. Gu, "Extreme learning machine-based classifier for fault diagnosis of rotating machinery using a residual network and continuous wavelet transform," *Measurement*, vol. 183, Oct. 2021, Art. no. 109864.

FAISAL ALTHOBIANI is currently an Associate Professor with the Marine Engineering Department, Faculty of Maritime, King Abdulaziz University, Jeddah, Saudi Arabia. He has more than six years of academic experience in teaching and research. He has authored several research papers in reputed journals, books, and conference proceedings. His current research interests include marine sciences, marine engineering, condition monitoring, and fault diagnosis systems.