



Research Article

An LSTM-Autoencoder Architecture for Anomaly Detection Applied on Compressors Audio Data

Pooyan Mobtahej ¹, Xulong Zhang,² Maryam Hamidi ¹ and Jing Zhang²

¹Department of Industrial Engineering, Lamar University, Beaumont, TX 77706, USA

²Department of Computer Science, Lamar University, Beaumont, TX 77706, USA

Correspondence should be addressed to Maryam Hamidi; mhamidi@lamar.edu

Received 30 December 2021; Revised 20 April 2022; Accepted 22 August 2022; Published 29 September 2022

Academic Editor: Yan Chai Hum

Copyright © 2022 Pooyan Mobtahej et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The compressors used in today's natural gas production industry have an essential role in maintaining the production line operational. Each of the compressors' components has routine maintenance tasks to avoid sudden failures. Hence, the significant advantages and benefits of performing preventative maintenance tasks in time are decreasing equipment downtime, saving additional costs, and improving the safety and reliability of the whole system. In this paper, anomaly classification and detection methods based on a neural network hybrid model named Long Short-Term Memory (LSTM)-Autoencoder (AE) is proposed to detect anomalies in sequence pattern of audio data, collected by multiple sound sensors deployed at different components of each compressor system for predictive maintenance. In research methodology, this paper has conducted experiments that employed different RNN architectures such as GRU, LSTM, Stacked LSTM, and Stacked GRU with various functions to create a baseline for model evaluation. Each architecture used audio signals dataset received from the compressor system for experiments to consider each neural network model's performance. According to performance results, an optimal model for anomaly detection with the best performance scores has been proposed in this research. Experiments combined one-dimensional raw audio signal features using SC and Mel spectrogram features were fed to deep learning models to evaluate performance. Hence, such hybrid methods can effectively detect normal and anomaly audio signals collected from a compressor system, increasing the compressor system's reliability and the sustainability of the gas production line. The combination of multiple-resource features in the proposed hybrid model showed a 100% score in all four-evaluation metrics such as accuracy, precision, recall, and F1 in LSTM-based autoencoder in both test and train results.

1. Introduction

Natural gas compressors are mainly used for providing pressure to transport gas in pipelines. The compression system interlocks upstream gas production and downstream consumer use by pressuring natural gas in channels. The compressors used in today's natural gas production industry have an essential role in maintaining the production line operational. Each compressor unit consists of multiple components and subsystems: an engine or electric motor, a compressor system, crankcase, valve body, one or numerous compression cylinders, cooling system, turbo unit, fan, and water pump. The existence of such components depends on the nature of the compressor. Each subsystem consists

of noise pollution contributed by the whole system of each compressor [1]. Each of these components has routine maintenance tasks to avoid sudden failures. Since the installation cost for a compressor is high, loss of the compression system can be costly due to repair costs and lost production economic effect [2]. Hence, the significant advantages and benefits of performing preventative maintenance tasks in time are decreasing equipment downtime, saving additional costs, and improving the safety and reliability of the whole system [3]. There are several approaches to conduct the preventative maintenance tasks in oil and gas organizations for compressor systems, such as time-based and periodic maintenance to check compressors in regular time intervals to perform troubleshooting and avoid potential failures.

However, predictive maintenance could be a solution since such components could also face unexpected or unscheduled downtime due to faults that negatively affect gas production. Such a method is based on predicting future failures of such components before occurrence. This approach works with real-time conditions of compressor systems and could reduce costs and unexpected downtime [2]. Each maintenance task is conducted for increasing production line reliability; hence, it is mandatory to keep the efficiency and functionality of gas production lines, which depends on the reliability of each of the components and systems, and compressors are part of it [4]. The technicians' original and current solution for predictive maintenance tasks consists of traditional approaches. It means that current predictive maintenance tasks use traditional human-based inspection methods and need experienced technicians to listen to compressor sounds and noises by the human ear to inspect the defects manually. Afterward, the technician would report required information of inspection and examination such as the timestamp of the event, the type of identified failure if found, and the timestamp of any conducted maintenance task. However, such a method consists of several disadvantages [5].

For instance, conducting such a traditional inspection method is very time-consuming, inaccurate, and challenging considering that the decibel level of a compressor sound could be louder than a jet engine, making it difficult for technicians to find defects properly. Additionally, the distance of reaching the component is usually far away from the reach of the technicians since the location of compressors and pumping stations is generally out of town. Hence, it needs to introduce an efficient automated approach for diagnosing defects and classifying normal abnormalities appropriately to solve such challenges [6, 7]. Therefore, this research paper aims to develop an automated system to accurately detect and classify normal audio signals and anomaly audio signals for midstream compressor systems. If such methods could classify and detect anomalies accurately, it would significantly reduce the risk of any sudden failure and loss in the production line at last.

Hence, this research paper has been organized as follows: Related studies for conducting the literature review and research contribution are mentioned in the next section. Basic definitions are reviewed in Section 3, methodology has been presented in Section 4, Section 5 shows the experimental results and discussion and the outcomes and would present the conclusions and future work suggested at last.

2. Related Studies

There is an in-depth literature review for the methods mentioned in similar studies based on statistical, machine learning-based supervised learning, and other deep learning-based unsupervised approaches for anomaly detection. For instance, there are several methods for such classifications, such as traditional statistical methods, which would be considered non-neural network methods. Those methods include K-means clustering, random forest, and machine learning and deep learning methods for anomaly detection

[6–8] and system prognostic [9], prediction [10], classification [11], and system reliability improvement [7]. For instance, random forest (RF) is a supervised learning algorithm in which several decision trees would be constructed while training is in progress [12]. RF would take the classes' mean to predict the trees as the output. RF would be used for nonlinear regression tasks [13]. As a research example, in Munir et al., several methods and modeling approaches have been mentioned to detect and classify anomalies, which is considered statistical modeling for conducting anomaly classification process [14]. For instance, K nearest neighbor (KNN) method is one of the most commonly used distance-based methods for anomaly detection. In KNN, for every data point in each dataset, K nearest neighbors would be considered to conduct classification tasks [14, 15]. In other words, according to these points, abnormalities in a sequence of time series data could be detected and classified. As presented in several works of literature, a histogram-based outlier score (HBOS) would also be used as a traditional anomaly detection method using an abnormality score for outlier detection [15, 16]. In this method, first, a histogram for the features would be generated, and then the data set instances would be multiplied to the inversed height of the bins of all features. HBOS is a statistically unsupervised method based on histograms. The advantage of using such a method is that it is less computationally expensive than distance-based and clustering-based anomaly detection [16]. In other research, such as Liu et al., isolation forest has been used for outlier detection. Isolation forest is based on density and distance measures. In other words, this approach would isolate normal instances from anomalies. This method would use a binary tree called isolation tree to conduct the anomaly detection task and isolate the abnormal instances [17]. Another method for anomaly detection would be extreme gradient boosting outlier detection (EGBOD). This method is semi-supervised learning for conducting anomaly detection tasks. Extreme gradient boosting is an extension of the gradient boosting algorithm method. This algorithm is defined to reduce the error caused by deviation from generalization [18]. This algorithm could be used for classification and regression modeling problems [13]. Another classification method is support vector machine. SVM was proposed as anomaly detection and classification method. SVM is the supervised learning method that has usage in classification models. For instance, one-class SVM is an unsupervised algorithm that aims to find the maximum margin hyperplane that best separates the data from the center [19]. Other traditional algorithms could be used for performing classification tasks effectively. Another example is the Naïve Bayes algorithm. Such an algorithm is basically could be used for supervised learning. Naïve Bayes [20] can solve classification problems. It could perform fast classification and quick predictions. Naïve Bayes would predict the probability of different classes based on various attributes which have been identified. A Naïve Bayes algorithm could be used in classification problems that consist of several classes [15, 20]. On the other hand, tree-based classification techniques such as decision trees and random forests [13, 21] are simple and very popular methods in data mining

problems. They can map nonlinear relationships effectively and efficiently [22, 23]. For instance, in the decision tree, each leaf node is assigned a label according to dataset classes. There are also nonterminal nodes, consisting of the root node and other internal nodes. They contain test conditions for separating instances with different characteristics [15, 23]. Both traditional machine learning and deep learning methods [6, 9] have advantages. However, according to this project specification, scope, and dataset, and because of high accuracy, efficiency, and reliability compared to traditional methods and other machine learning algorithms, deep learning-based anomaly detection methods have been proposed in this research. Furthermore, deep learning technology has grown significantly in the past few years and has achieved many impressive impacts in various research areas. Such areas include artificial intelligence, computer vision, deep medicine [24], pattern recognition, predictive modeling [25], and image and signal processing [26], which are widely used in different areas such as healthcare, energy industry, automation, and manufacturing. Various deep learning and machine learning models have been used to conduct similar tasks in similar research studies for anomaly detection and classification. For example, Kavitha et al. used support vector regression (SVR) for classification. SVR would be used for supervised learning. The problems which SVR could address are regarding classification and regression. The process would be like a support vector machine (SVM) with some changes. In SVR, the appropriate line for fitting the model would be defined based on acceptable predefined residuals [27]. In another example from the deep learning-based model for anomaly detection, DeepAnT is proposed [14, 28]. This model is an unsupervised anomaly detection method used in time series data. This method consists of two modules. The first module is a predictor for the time series data, while the value instances are fed into the anomaly detector module. This module could detect normal and anomalies. The predictor module is based on a convolutional neural network (CNN). Additional to that, FuseAD was introduced based on DNN and statistical methods for unsupervised anomaly detection consisting of two modules. The first module is the forecasting modules used ARIMA [29], and the forecast output would feed into the CNN module along with time series data. CNN is appropriate for analyzing imagery data and has high capability in the feature extraction process [30]. The output of these modules would be augmented by a summation layer at last [14, 28]. ARIMA model, which has been used in the mentioned method, is used to forecast and analyze time series data in linear form. The autoregressive integrated moving average model is an extended model from the ARMA that would use lagged observation to predict observation in time series data. ARIMA [31] is one of the most effective models in machine learning for forecasting time series data and basically would perform regression in previous time steps to predict the next instance [25].

As mentioned regarding CNN models, before applying the proposed method in this research phase, ResNet50 has been used in two datasets collected from compressors as the first phase of this project published previously [30, 32].

In the mentioned study, the ResNet50 convolutional neural network model extracted high-level features from the input data [30, 33]. ResNet50 consists of 50 different layers, including different layers such as max pooling, convolution, and fully connected layers. A ResNet50 is defined as a pre-trained network that used ImageNet database. This database consists of more than 14 million images in a wide range of categories [34, 35]. In the CNN architecture which has used ResNet50 as a deep learning layer [34], Mel-frequency Cepstral coefficients (MFCC) have been computed using the input audio signals which created a two-dimensional matrix of features. On the other hand, SC features are obtained from the input audio signals. As the next step, for extracting high-level deep features from MFCC, a pretrained deep learning neural network based on ResNet50 was used. Then, both deep MFCC features and SC features are fed to a principal component analysis (PCA) unit for the final step as feature extraction and reducing the dimensions. The extracted MFCC and SC features are combined to train a support vector machine (SVM) classifier in which could conduct normal and anomaly audio signal classification tasks. As mentioned in this study, MFCC [36] would be defined as one of the familiar data representations from the audio signal which could be used for further processing and feature extraction [30, 32]. In such a feature extraction method, the spectrum of the input audio signal after fast Fourier transform (FFT) is going to be filtered by Mel filters. This would create Mel spectrum, and as the next step, the Cepstral analysis is carried out to the logarithm of the Mel spectrum. At last, Cepstral coefficients earned by the mentioned process are defined as Mel-frequency Cepstral coefficients (MFCC) [37, 38]. Additional to that, PCA has also been mentioned as part of the method. PCA would be used in data compression and dimension reduction. PCA process could keep important information of the data and can convert the data from a higher-dimensional space into a lower-dimensional space. This process could reduce the size and dimensions. A 2-fold cross-validation used to evaluate the performance using datasets for ResNet50 architecture [28, 30]. The mentioned model has been summarized in Figure 1 [30].

In the conducted experiments, two datasets consisting of 2343 raw audio signals in dataset 1 with 156 anomaly audio signals in the OGG format and 7853 audio signals with 1085 anomaly audio signals in dataset 2 have been used for the classification and detection task experiments. Both datasets were very unbalanced. There are many more normal samples than anomaly samples in the dataset. The output of the PCA feature set contains combined deep MFCC features extracted by the ResNet50 model and the SC features. This feature set was then fed to train SVM classifier to classify the audio signals into normal and anomaly classes as mentioned in the method. In the mentioned research [30, 32], experiments demonstrated that combined deep MFCC features and SC features achieved the best performance for normal and anomalies classification for all four-evaluation metrics used in the experiments named precision, recall, accuracy, and F1 [39].

Along with using the mentioned methods and models in anomaly detection, there are also hybrid models based on

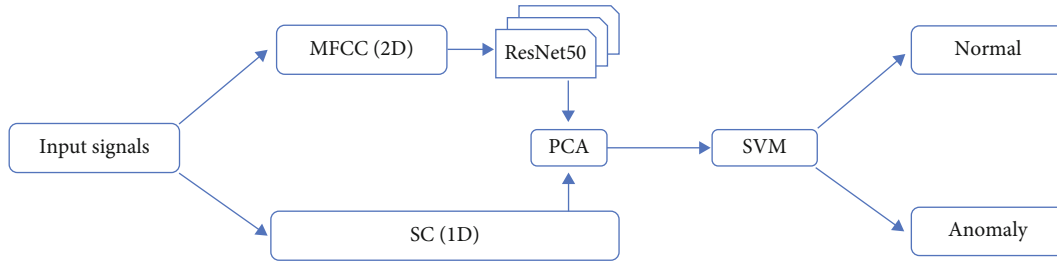


FIGURE 1: The deep learning architecture based on ResNet50 for anomaly detection [30].

deep learning for similar research questions. For example, the CNN-LSTM [40, 41] is one of the most used hybrid machine learning models to increase performance, and the capability to make better predictions. LSTM-based model [42] could extract required information in sequence patterns and time series data by its gated architecture [43]. On the other hand, CNN can extract important features by reducing noise and filtering algorithms [30, 44]. Additionally, Convolutional LSTM is a sequential model, based on LSTM. Its internal layers are replaced with convolutional layers and would be able to extract the best features to feed the classifier for better performance in the classification task. As another example, stacked autoencoder (SAE) has been proposed in similar research for unsupervised learning. SAE includes layers such as input, output, and hidden layers in its neural network architecture. In SAE, encoder and decoder would conduct the process of training. In this model, the main steps are including training the input data by autoencoder and feeding the next layer for training until training of the network would be completed, and at last, backpropagation algorithm would be used for minimizing the cost function. Then, layers weights would be updated by training set to conduct fine tuning [45]. The advantages of the hybrid models in neural networks are the capability of such models to overcome the shortcomings of massive and high-dimensional datasets [46]. According to considered similar literature, the hybrid deep learning neural network model to classify and detect anomalies could be an appropriate model and a baseline for predicting mechanical component failures in this paper [9].

2.1. Research Contribution. In comparison to similar literature mentioned, in this research project, sound sensors (e.g., smart microphones) have been widely used in the time series data collection process. Multiple sound sensors are deployed at different components of a compressor system which can be controlled by the Internet of Things (IoT) techniques to collect and transfer the auditory data in a specific period for storing in a database. As the next step, advanced data processing and mentioned deep learning model have been used for audio classification of normal and anomaly signals. The defined deep learning network model needs to be trained on each individual component using normal and abnormal sound signals for anomaly detection and classification [2, 6]. Therefore, in this paper, the main contributions are mentioned as the following points:

- (i) This study proposes a different hybrid neural network method for applying to collected audio signals to detect anomalies and classify them accurately. Therefore, according to the research problem mentioned in this paper, LSTM-Autoencoder architecture [45, 46] is designed to analyze one-dimensional raw audio signals. In another word, audio signal features consisting of Mel spectrogram and SC features would be extracted from raw data for feeding the network to be trained and using that for anomaly detection in test data to evaluate the performance of the proposed method as well as comparing that with other architectures [31, 32]. Audio raw data collected from compressors audio sensors would be used to conduct experiments on the mentioned model to see whether they could classify the normal and anomaly effectively by using appropriate functions, configuration, and classification method and presenting the result at last and evaluate it with validation dataset to assure that the model is well fitted without any overfitting [17, 33]
- (ii) The proposed hybrid model has been compared with baseline models such as GRU, Stacked GRU, LSTM, and Stacked LSTM to assure that the model performs better than other RNN-based models with different hyperparameters and functions in the same experiments
- (iii) This study highlights the importance of this research area and bolds the importance of the application of deep learning methods in anomaly detection for time series data in predictive maintenance (PdM). PdM would reduce maintenance costs and to have sustainable operational components such as compressors. The core of PdM is to predict the next failure which could lead to conducting maintenance tasks that can be scheduled before it happens [31]. Such methods are basically based on statistical calculation for estimating the time to failure based on the maintenance data; however, this paper proposed deep learning-based approach which is more accurate
- (iv) In the previous study mentioned in related studies section for anomaly detection, we have used ResNet-50 with pretrained network and employed

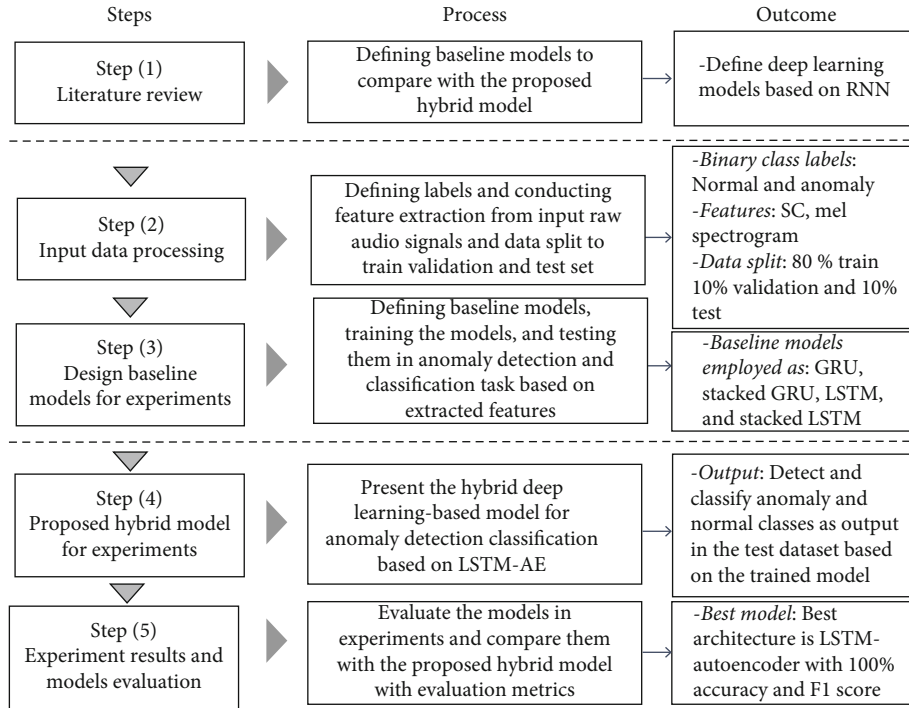


FIGURE 2: Comprehensive methodology flowchart: represent step by step methodology process, the architectures of baseline and proposed hybrid model, the conducted tasks on input data and output, and the experiment result (created by authors).

MFCC and SC as feature sets for anomaly detection with smaller datasets [30]. This research is a complementary study in which has in-depth study in different RNN variant architectures as baseline models and proposed a hybrid LSTM-AE for anomaly detection as best-scored model at last. This study used bigger dataset and employed Mel spectrogram and SC as feature sets for training the model to evaluate the performance of different RNN models, hyperparameters, and layers in anomaly detection and classification of unseen data using audio signals. Additionally in the previous study, the MFCC features have been used in ResNet50 and then combined with SC feature. PCA applied for dimensionality reduction and SVM employed for classification task. However, in this study, both Mel Spectrogram and SC features fed directly to the network, and in LSTM-AE, the dimension reduction and information gain have been processed within the network and last layer with Soft-Max layer conduct the classification task

- (v) In this study, a hybrid deep learning architecture model based on LSTM-AE has been proposed for the analysis of operational data such as audio signals from the working compressor which could be the basis for increasing the reliability of operational components of compressor systems in a more efficient approach. Such methods could be more accurate, faster in analysis, and more effective for having a more sustainable production line according to performance analysis results in this paper [37, 47].

Therefore, the importance of the application of using the deep learning method could be highlighted in this research rather than using other similar approaches. Time and computational cost analysis have been presented as well for the proposed method

3. Definitions and Notations

This section would describe the terms, features, and functions that have been used in the paper.

3.1. Features: Spectral Centroid and Mel Spectrogram. Spectral centroid (SC) [38] is regarding the brightness of the sound. In another word, SC would show where the midpoint of mass of the spectrum is in the audio signal [48]. In the other hand, in *Mel spectrogram*, the frequency of the signal would be converted to log scale and amplitudes which could show the spectrogram. Afterward, the frequency domain would be mapped to the Mel scale to shape the Mel spectrogram at last [49, 50].

3.2. Recurrent Neural Network (RNN). Applying deep learning methods on time series data has been widely used in many areas such as fault detection in manufacturing machines [51, 52]. RNN has a neural network architecture in which it would use time series data for prediction and remembering important information. There are some most used and common RNN cells such as LSTM and GRU [47, 53]. GRU is very similar to LSTM architecture with less complexity and computational time [53, 54].

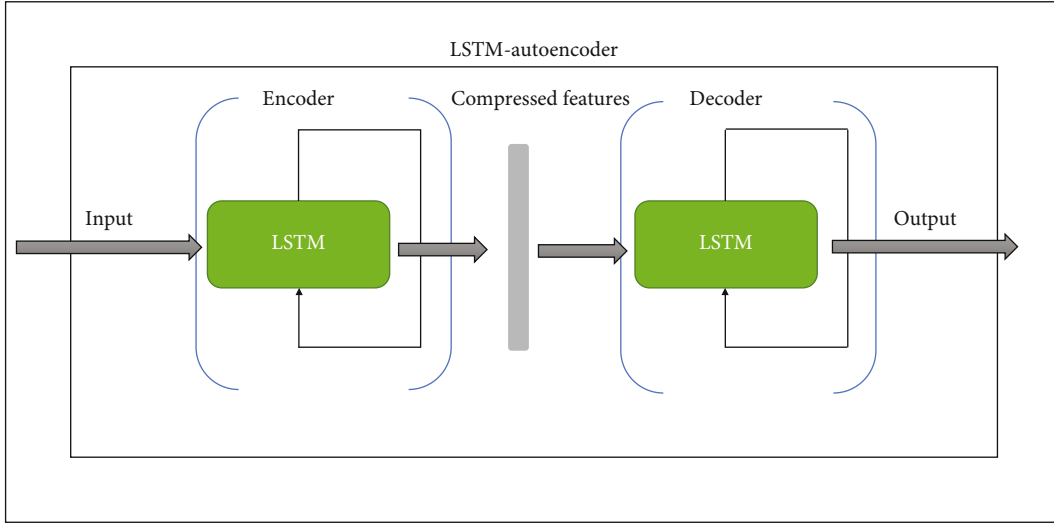


FIGURE 3: LSTM-AE model: the architecture shows the structure of the network for encoding and decoding the input sequence for classification (created by authors).

3.3. *Deep Learning Models' Functions.* Different functions and layers in the model have been used in this paper for achieving better performance and tuning [55].

3.4. *Activation Functions.* Sigmoid function has the sigmoid curve characteristic where $g(x)$ is the Sigmoid function and e is Euler's number and the output would be between 0 to 1 which makes it a good choice for prediction [56]:

$$g(x) = \frac{1}{1 + e^{-x}}. \quad (1)$$

SoftMax function would generalize the logistic regression which would be used in classification task for multiple classes where S is softmax, $z \rightarrow$ is defined as input vector and e^{z_i} is standard exponential function applying on the input vector, n is also defined as a number of classes, and e^{z_j} is a standard exponential function which performed on output vector [57, 58]:

$$S\left(\vec{z}\right)_i = \frac{\exp^{z_i}}{\sum_{j=1}^n \exp^{z_j}}. \quad (2)$$

3.5. *Loss Functions.* Categorical cross entropy would be used as loss function in conducting classification in multiple classes where W is the model parameters such as weights of the neural network, n is defined as the number of data points, y_i is true labels, and p_i is the predicted labels [59, 60]:

$$L(w) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]. \quad (3)$$

The difference between categorical cross entropy and sparse categorical cross entropy [57] loss functions is regarding the format of true labels. For instance, if you use integers in true labels, we can use sparse categorical entropy. Additionally, mean squared error (MSE) would be able to mea-

sure the average of the squares of the errors where among data points in true and predicted labels [61].

4. Methodology

In this section, the proposed method for conducting classification and detection tasks as well as baseline methods has been presented for giving an overview of the performance of the experiments. The suggested method is based on LSTM-Autoencoders, and baseline methods are based on other recurrent neural networks such as GRU, LSTM, Stacked LSTM, and Stacked GRU which have similar core architecture as the proposed method to show the qualification and efficiency of LSTM-based autoencoder. Figure 2 shows the step-by-step process in the methodology. The methodology section is then followed by a description of the proposed hybrid model and baseline models architecture in detail.

4.1. *LSTM-Based Autoencoders' Anomaly Detection.* LSTM-based autoencoder is a neural network architecture that uses the autoencoder architecture for encoding and decoding a sequence of input data. The encoder-decoder process performs by the LSTM network in LSTM-AE. For a better understanding of the model, we would describe each part of the model such as autoencoder and LSTM separately as follows.

4.2. *Long Short-Term Memory (LSTM).* RNN cannot memorize important information from a sequence of data; therefore, an extension of RNN named LSTM could be used. Such network architecture could process an entire sequence of data. Such architecture could have longer memory rather than RNN. LSTM consists of feedback connections which mean the entire sequence data, as well as a single data point, could be processed by the network. And each cell unit of the network could be updated each time. LSTM has been used for part of the experiments of audio data anomaly detection

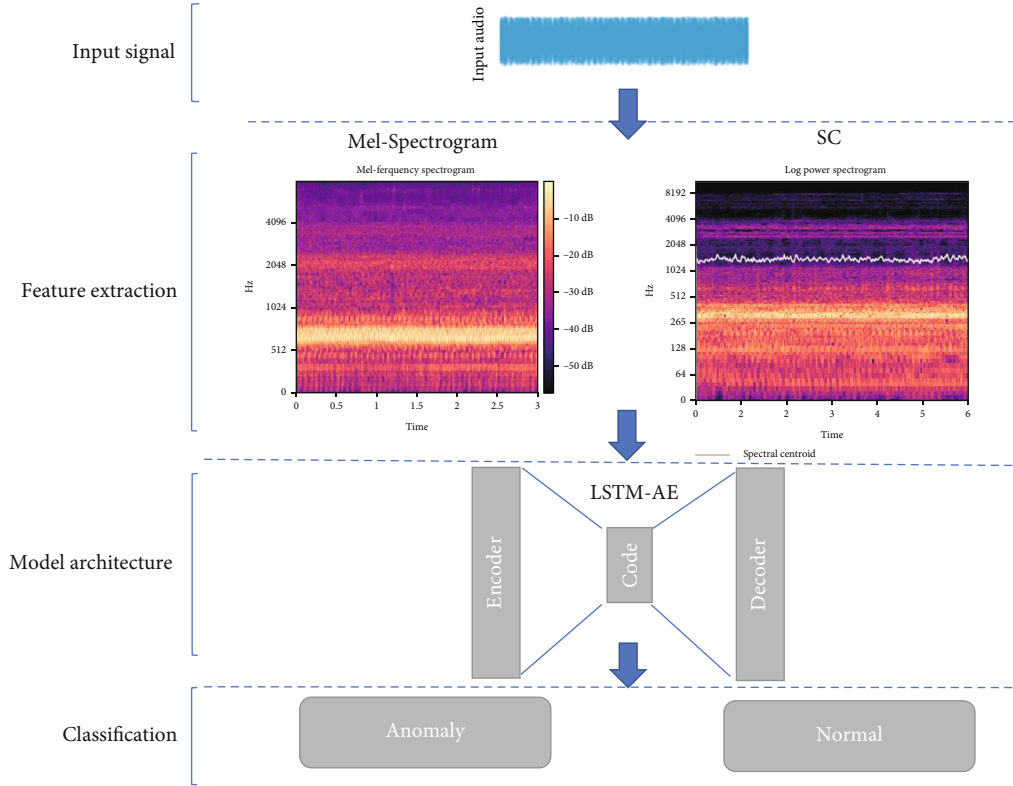


FIGURE 4: Anomaly detection and classification architecture using LSTM-AE: the figure shows the features' encode and decode process for the 2-class classification of detected anomalies (created by authors).

and classification in this research. As it has been mentioned, each cell could remember the past sequence of data and combine the information with the current input sequence. LSTM cell would consist of three gates which are named input gate, output gate, and forget gate. In these gates, the functionality of the network has mathematical representations which are presented in the following equations.

An LSTM network input gate would be responsible for deciding about the information which needs to be transferred to the cell. The following equations which have been defined previously and in related studies would describe its mathematical process starts with describing the input gate [53]:

$$i_t = \sigma(W_i * [h_{t-1}, x_t]) + b_i. \quad (4)$$

Forget gate would decide about keeping the required information from the previous memory state. The mathematical equation of this process has been mentioned as follows:

$$f_t = \sigma(W_f * [h_{t-1}, x_t]) + b_f. \quad (5)$$

The information in each cell would be updated by update gate in which the following equations show its mathematical process:

$$\tilde{C}_t = \tanh(W_c * [h_{t-1}, x_t]) + b_c, \quad (6)$$

$$C_t = f_t * c_{t-1} + i_t * \tilde{C}_t. \quad (7)$$

The previous time step in the hidden layer would be updated by the output gate. This gate would also update the output of the given data as well. The following equations represent the process:

$$o_t = \sigma(W_o * [h_{t-1}, x_t]) + b_o, \quad (8)$$

$$h_t = o_t * \tanh(c_t), \quad (9)$$

where x_t represents the input of the cell, while h_{t-1} , h_t and c_{t-1} , c_t would describe the hidden states and the cell states. The rest of the mentioned variables would describe trainable weights and biases which would be used in the LSTM model [47, 53].

4.3. Autoencoder. AE consists of two modules: encoder and decoder. The data would be fed to the encoder which would learn the underlying features of a process. These features are basically in a reduced dimension. The decoder on the other hand would recreate the original data from these underlying features. The output would have reduced noise and dimensions. Basically, an autoencoder could effectively compress features from high-dimensional data. Employing AE is very popular in anomaly detection problems since AE could significantly increase the accuracy of abnormalities detection which other methods such as PCA would fail to conduct [39]. Another advantage of using AEs is that it is easy to train and does not require computational complexity which other methods like kernel PCA face. Autoencoder

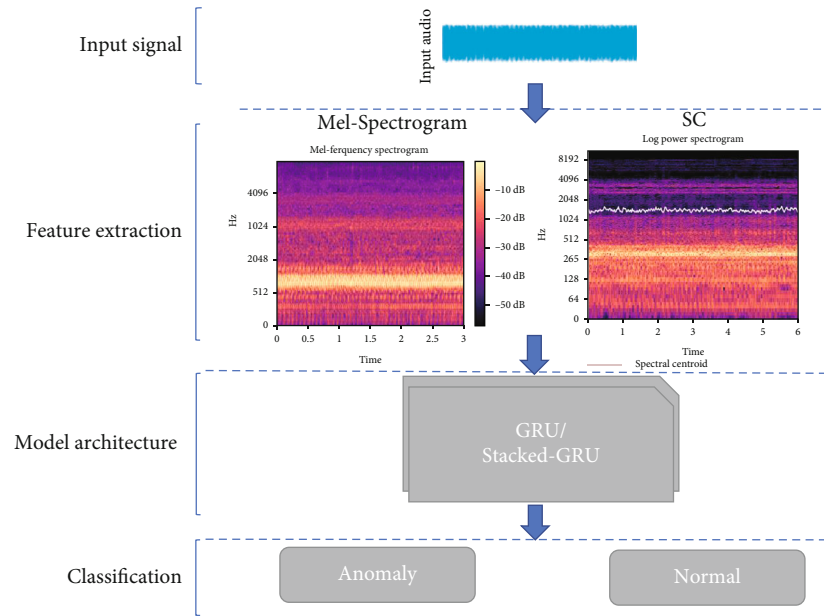


FIGURE 5: Anomaly detection architecture using GRU/Stacked GRU: the figure specifies the process of employing GRU/Stacked GRU architecture for conducting the anomaly detection and classification task (created by authors).

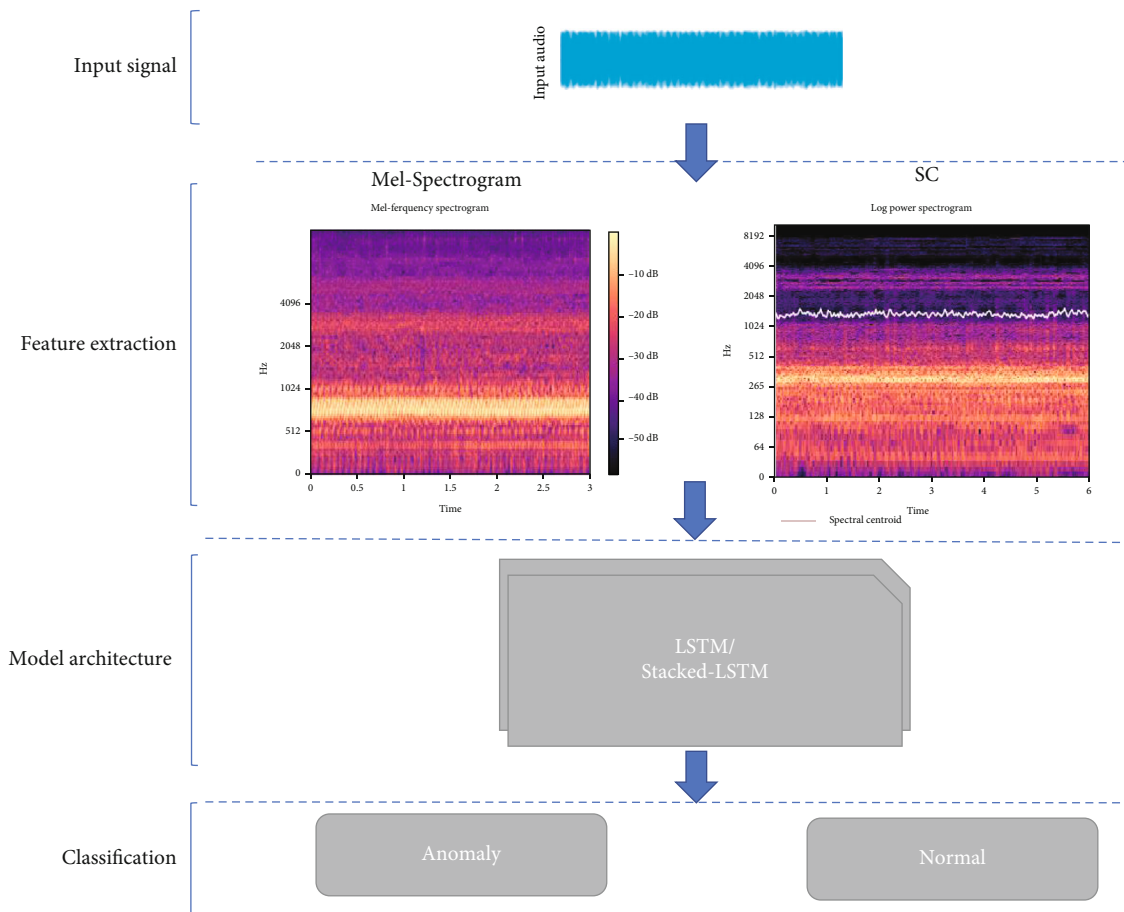


FIGURE 6: Anomaly detection architecture using LSTM/Stacked LSTM: the figure specifies the process of employing LSTM/Stacked LSTM architecture for conducting the anomaly detection and classification task (created by authors).

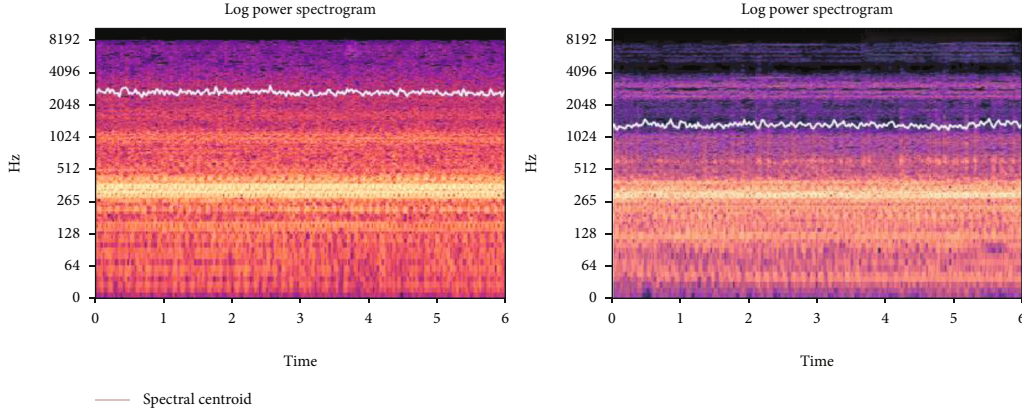


FIGURE 7: SC data representation for normal and anomaly: The figure would show the difference between normal and anomalies in using the SC feature (created by authors).

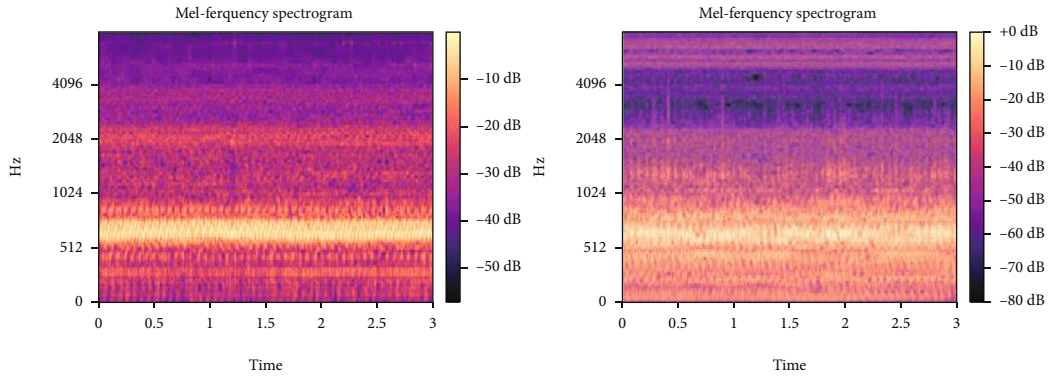


FIGURE 8: Mel spectrogram data representation for normal and anomaly: The figure would show the difference between normal and anomalies in using the Mel spectrogram feature (created by authors).

architecture could handle large-scale data and conduct feature selection which made it an appropriate choice for applying to this research work domain [46, 62].

4.4. LSTM-Autoencoder. Based on mentioned advantages for LSTM and autoencoders, therefore, LSTM-based autoencoders [46, 63] are defined as a main proposed hybrid model which could effectively conduct feature selection among fed features based on information importance, anomaly detection, and classification task in audio data. Figure 3 gives an overview of the defined hybrid deep neural network architecture [15].

To clarify the process, original one-dimensional auditory data would be used for extracting spectrum centroid (SC) features to feed the model as input to the LSTM-based AE for training, which can maintain information in the state temporarily for a long number of time steps. Mel spectrogram features created using auditory data would be also used as feed input to the LSTM-AE. Such features have been selected based on using combined deep features in similar research which showed the best performance in the same work domain for anomaly detection [30]. The semi-supervised training will be applied to train the model. The process steps overview would be mentioned as Figure 4.

TABLE 1: Model parameters for LSTM-AE architecture (created by authors).

Model hyperparameters and configuration	
Model architecture	LSTM-Autoencoder
Number of classes	2
Number of layers	8
Batch size	128
Number of epochs	50
Number of units	50
Dropout rate	0.1
Total number of trainable parameters	93,004

4.5. Anomaly Detection Using GRU and Stacked GRU. GRU is the extended model from RNN which is less complex than LSTM and consists of two gates for using the information in a gated recurrent unit. To train the GRU, original one-dimensional auditory data would be used for extracting spectrum centroid (SC) features to feed the model as input to the GRU. On the other side, Mel spectrogram features created using auditory data would be also used as input to the GRU layers. The semi-supervised training will be applied

TABLE 2: Model dimensions and operations overview for LSTM-AE architecture (created by authors).

Operation		Data	Dimensions	Weights (N)	Weights (%)
Input	#####	2	259		
LSTM	LLLLL	—	—	62,000	66.7%
tanh	#####		50		
Dropout		—	—	0	0.0%
	#####		50		
Repeat vector		—	—	0	0.0%
	#####	259	50		
LSTM	LLLLL	—	—	20,200	21.7%
tanh	#####	259	50		
Dense		—	—	0	0.0%
	#####	259	50		
Time distributed		—	—	102	0.1. %
	#####	259	2		
LSTM	LLLLL	—	—	10,600	11.4%
tanh	#####		50		
Dense	XXXXXX	—	—	102	0.1%
SoftMax	#####		2		

TABLE 3: Model parameters and configurations for GRU network (created by authors).

Model Hyperparameters and configuration	
Model architecture	GRU
Number of classes	2
Number of layers	3
Batch size	128
Number of epochs	50
Number of units	50
Dropout rate	0.1
Total number of trainable parameters	46,752

TABLE 4: Model dimensions and operations for GRU employing Sigmoid layer (created by authors).

Operation		Data	Dimensions	Weights (N)	Weights (%)
Input	#####	2	259		
GRU	LLLLL	—	—	46,650	99.8%
tanh	#####		50		
Dropout		—	—	0	0.0%
	#####		50		
Dense	XXXXXX	—	—	102	0.2%
Sigmoid	#####		2		

to train the model. The same process would be applied to conduct experiments in Stacked GRU model architecture. In a stacked GRU, there are multiple GRU layers which are called stacking. The main reason for defining such a

model is the ability to conduct greater complexity in experiments, and increasing the layer numbers would increase the ability to conduct anomaly detection and classification more accurately. The process overview would be mentioned as Figure 5.

4.6. Anomaly Detection Using LSTM and Stacked LSTM. To train the LSTM model in this model, original one-dimensional auditory data would be used for extracting spectrum centroid (SC) features to be used as input to the LSTM model which can maintain temporal information from data in each cell for a longer number of time steps and has three gates and more complexity than GRU networks. Mel spectrogram features that have been extracted by using auditory data would be also used as feed input to the LSTM network. The semi-supervised training will be applied to train the model. The process would be also applied to the Stacked LSTM design as well. Stacked LSTM models are deep networks with multiple LSTM hidden layers which have been connected. In such model architecture, the output of an LSTM hidden layer will be fed into the next of the LSTM layer as the input. These stacked layers design would improve the learning capability of neural networks and fit the model performance more effectively [64]. The process for both LSTM and Stacked LSTM is mentioned in Figure 6.

5. Experimental Study

According to proposed deep learning methodologies, this study has used a dataset from the Turbo component of the selected compressor. Turbo is one of the most important components of each compressor system, and its full functionality would be crucial for the production line to be operational. This section would explain the details of the dataset which has been

TABLE 5: Model dimensions and operations for GRU network employing SoftMax layer (created by authors).

Operation	Data	Dimensions	Weights (N)	Weights (%)
Input	#####	2	259	
GRU	LLLLL	—	—	46,650
tanh	#####		50	
Dropout		—	—	0
	#####		50	0.0%
Dense	XXXXX	—	—	102
SoftMax	#####		2	0.2%

TABLE 6: Model parameters for Stacked GRU architecture (created by authors).

Model Hyperparameters and configuration	
Model architecture	Stacked GRU
Number of classes	2
Number of layers	6
Batch size	128
Number of epochs	50
Number of units	50
Dropout rate	0.1
Total number of trainable parameters	77,352

used in experiments on different mentioned models in methodology as well as each experiment specification.

5.1. *Experimental Data.* The dataset specification and details which have been used in this research are listed as follows.

5.2. *Dataset's Specification.* The following dataset collected by Well Checked Systems International from the Turbo component of the selected compressor system has been used for experiments in the proposed anomaly detection methodology.

- (i) *Dataset:* This dataset has a total of 12,190 raw audio signals collected from the Turbo component system of the selected compressor in a specific period as time series data with two class labels as normal and anomalies. There are 8559 normal audio signals and 3631 anomaly audio signals. Each signal is 3 seconds long and saved in the OGG format. This is an unbalanced dataset which means that there are many more normal samples than anomaly samples

The normal data has been defined based on the normal working conditions of the Turbo component of the compressor system. The sounds of normal condition of the Turbo component have been used for labeling the normal class. However, on the other hand, the anomalies were the cases that consist of a noise in the working condition audio signals. In this research, two classes of normal and anomaly are defined to detect and classify anomalies; however, the fault reasons and root of the anomaly could be checked by technicians during conducting maintenance task.

TABLE 7: Model dimensions and operations overview based on Stacked GRU (created by authors).

Operation	Data	Dimensions	Weights (N)	Weights (%)
Input	#####	2	259	
GRU	LLLLL	—	—	46,650
tanh	#####	2	50	
Dropout		—	—	0
	#####	2	50	0.0%
GRU	LLLLL	—	—	15,300
tanh	#####	2	50	
Dropout		—	—	0
	#####	2	50	0.0%
GRU	LLLLL	—	—	15,300
tanh	#####		50	
Dropout		—	—	0
	#####		50	0.0%
Dense	XXXXX	—	—	102
SoftMax	#####		2	0.1%

TABLE 8: LSTM model parameters and configurations (created by authors).

Model Hyperparameters and configuration	
Model architecture	LSTM
Number of classes	2
Number of layers	3
Batch size	128
Number of epochs	50
Number of units	50
Dropout rate	0.1
Total number of trainable parameters	62,102

To give an overview of data feature representation, the following figures were created to show the anomaly and normal difference in two samples of audio signals in each class. Hence, the selected features represented as SC and Mel Spectrogram are as follows.

5.3. *Spectral Centroid Representation.* The data representation as a generated image for normal and anomaly sample data from original audio signals for SC is mentioned in Figure 7. The difference between normal and anomaly signals is clear in the log power spectrum and SC representation.

5.4. *Mel Spectrogram Representation.* The data representation as an image for normal and anomaly samples from the original audio dataset for Mel Spectrogram differences in Mel frequency spectrogram is shown in Figure 8.

TABLE 9: Model dimensions and operations overview for LSTM employing Sigmoid layer (created by authors).

Operation		Data	Dimensions	Weights (N)	Weights (%)
Input	#####	2	259		
LSTM	LLLLL	—	—	62,000	99.8%
tanh	#####		50		
Dropout		—	—	0	0.0%
	#####		50		
Dense	XXXXX	—	—	102	0.2%
Sigmoid	#####		2		

5.5. *Experiments.* Experiment specification and detailed view of steps in the proposed method as well as baseline models have been mentioned in this section.

5.6. *Experiment Specification Using LSTM Auto-Encoder.* In this research, the network would be trained by 80% of the dataset consisting of both normal and anomaly audio signal extracted features. The remaining data were used for validation and testing. Each of them consists of 10% of the remained data in the shuffled form. SC and Mel spectrogram features have been extracted and have 259 dimensions to feed the LSTM-Autoencoder. In this research, the network consists of 50 units. The designed network input shape consists of several units, several features, and data dimensions.

SoftMax [57] and Sigmoid function have been used as activation functions in the output layer in different experiments. Such functions would work based on the probability distribution of the output to predict the result label. This function has been used for the detection of normal and anomalies, and the final used confusion matrix would show the performance of the network in the test and train dataset at last.

However, to create an optimal model, different functions such as sparse categorical cross entropy for loss function and various activation functions have been tested, and the Adam optimization algorithm for training the deep learning model has been employed to change the required attributes for reducing the losses of the model. The result of experiments on train and test datasets has been presented in the next section.

There are several approaches to validate the model such as splitting the dataset into train, test, and validation. By considering the model loss and accuracy in the train and validation dataset, the model could be evaluated against overfitting and underfitting. The dropout technique has been also employed for avoiding overfitting by dropping out some of the units in the network by a defined rate.

The plots for each experiment’s accuracy and loss and interpretation have been added to each result section for the LSTM model. The audio signal classification and anomaly detection using the LSTM-AE network program was implemented using Python. Additional to that, Keras, NumPy, Pandas, and TensorFlow library packages have been used for implementing the program and building the neural network model. The experiments have been con-

TABLE 10: Model dimensions’ overview and operations for LSTM employing SoftMax layer (created by authors).

Operation		Data	Dimensions	Weights (N)	Weights (%)
Input	#####	2	259		
LSTM	LLLLL	—	—	62,000	99.8%
tanh	#####		50		
Dropout		—	—	0	0.0%
	#####		50		
Dense	XXXXX	—	—	102	0.2%
SoftMax	#####		2		

TABLE 11: Stacked LSTM model parameters and configuration (created by authors).

Model hyperparameters and configuration	
Model architecture	Stacked LSTM
Number of classes	2
Number of layers	6
Batch size	128
Number of epochs	50
Number of units	50
Dropout rate	0.1
Total number of trainable parameters	102,502

TABLE 12: Model dimensions and operations view for stacked LSTM employing SoftMax layer (created by authors).

Operation		Data	Dimensions	Weights (N)	Weights (%)
Input	#####	2	259		
LSTM	LLLLL	—	—	62,200	60.5%
tanh	#####	2	50		
Dropout		—	—	0	0.0%
	#####	2	50		
LSTM	LLLLL	—	—	20,200	19.7%
tanh	#####	2	50		
Dropout		—	—	0	0.0%
	#####	2	50		
LSTM	LLLLL	—	—	20,200	19.7%
tanh	#####		50		
Dropout		—	—	0	0.0%
	#####		50		
Dense	XXXXX	—	—	102	0.1%
SoftMax	#####		2		

ducted in a MacBook Pro laptop with 2.4 GHz 8-Core Intel Core i9, 32 GB of Memory, and AMD Radeon Pro 5500 M 4 GB for experiments processing.

The model hyperparameters and configuration, as well as model operations and dimensions detailed view, have been mentioned in Tables 1 and 2 for the conducted experiments.

TABLE 13: Performance results on train dataset (created by authors).

<i>Model architecture</i>	<i>Loss function</i>	<i>Activation Function</i>	<i>Optimizer</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
GRU	Mean square error	SoftMax	Adam	89%	88%	55%	67%
GRU-1	Sparse categorical cross entropy	Sigmoid	Adam	99%	100%	99%	99%
GRU-2	Sparse categorical cross entropy	SoftMax	Adam	99%	99%	99%	99%
Stacked GRU	Sparse categorical cross entropy	SoftMax	Adam	99%	99%	99%	99%
LSTM	Sparse categorical cross entropy	Sigmoid	Adam	99%	100%	97%	98%
LSTM-1	Sparse categorical cross entropy	SoftMax	Adam	99%	100%	99%	99%
Stacked LSTM	Sparse categorical cross entropy	SoftMax	Adam	99%	100%	99%	99%
LSTM- Autoencoder	Sparse categorical cross entropy	SoftMax	Adam	100%	100%	100%	100%

TABLE 14: LSTM-AE result on test dataset performance with baseline model comparison (created by authors).

<i>Model architecture</i>	<i>Loss function</i>	<i>Activation function</i>	<i>Optimizer</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
GRU	Mean square error	SoftMax	Adam	87%	87%	52%	65%
GRU-1	Sparse categorical cross entropy	Sigmoid	Adam	99%	100%	98%	99%
GRU-2	Sparse categorical cross entropy	SoftMax	Adam	99%	99%	99%	99%
Stacked GRU	Sparse categorical cross entropy	SoftMax	Adam	99%	98%	100%	99%
LSTM	Sparse categorical cross entropy	Sigmoid	Adam	97%	100%	97%	98%
LSTM-1	Sparse categorical cross entropy	SoftMax	Adam	99%	100%	98%	99%
Stacked LSTM	Sparse categorical cross entropy	SoftMax	Adam	99%	100%	99%	99%
LSTM- Autoencoder	Sparse categorical cross entropy	SoftMax	Adam	100%	100%	100%	100%

5.7. *Experiment Specification Using GRU.* The method used extracted Mel spectrogram features and SC features to train GRU network, and different loss functions such as mean square error and activation function such as sigmoid and SoftMax have been tested for this model. To be able to create a baseline to show the efficiency of the proposed hybrid model based on LSTM-AE, the experiment specification process would remain the same as mentioned for LSTM-AE. The model hyperparameters and configuration, as well as model operations and dimension detailed view, has been mentioned in Tables 3–5.

5.8. *Experiment Specification Using Stacked GRU.* The method used extracted Mel spectrogram features and SC features to train Stacked GRU to be able to create a baseline to show the efficiency of proposed hybrid model based on LSTM-AE; the experiment specification process would be remained the same as mentioned for LSTM-AE. The model hyperparameters and configuration as well as model operations and dimension detailed view have been mentioned in Tables 6 and 7.

5.9. *Experiment Specification Using LSTM.* The method used extracted Mel spectrogram features and SC features to train the LSTM network. To be able to create a baseline to show the efficiency of the proposed hybrid model based on LSTM-AE, the experiment specification process would remain the same as mentioned for LSTM-AE. The model hyperparameters and configuration, as well as model operations and dimension detailed view, have been mentioned in Tables 8–10.

5.10. *Experiment Specification Using Stacked LSTM.* The method used extracted Mel spectrogram features and SC features to train Stacked LSTM network. To be able to create a baseline to show the efficiency of the proposed hybrid model based on LSTM-AE, the experiment specification process would remain the same as mentioned for LSTM-AE.

The model hyperparameters and configuration as well as model operations and dimension detailed view have been mentioned in Tables 11 and 12.

5.11. *Evaluation Metrics.* Four evaluation metrics have been defined for evaluating the anomaly detection performance in each of the experiments as well as the proposed LSTM-Autoencoder network for anomaly classification and detection in train and test experiments. The accuracy, precision, recall, and F1 have been used as evaluation metrics [65] which have been defined in the following equations [47]:

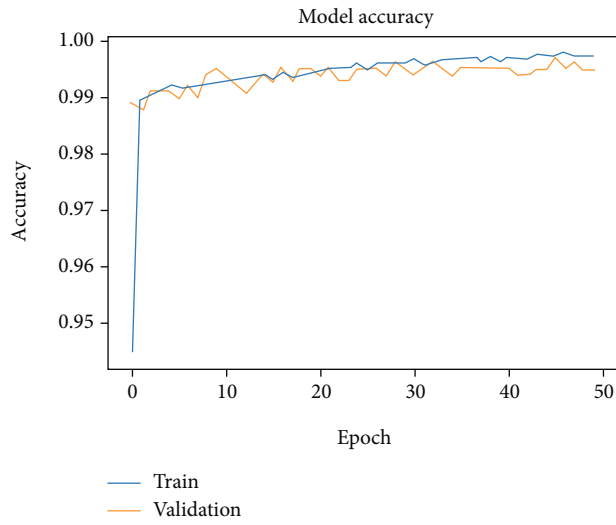
$$Accuracy = \frac{TP + TN}{Total\ Number\ of\ Samples}, \quad (10)$$

$$Precision = \frac{TP}{TP + FP}, \quad (11)$$

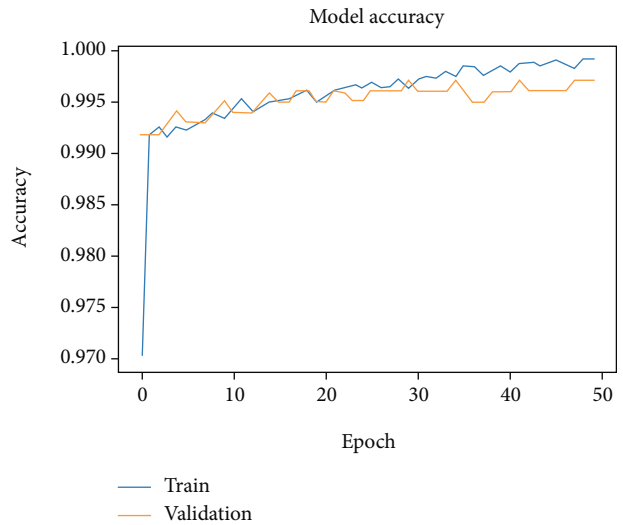
$$Recall = \frac{TP}{TP + FN}, \quad (12)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (13)$$

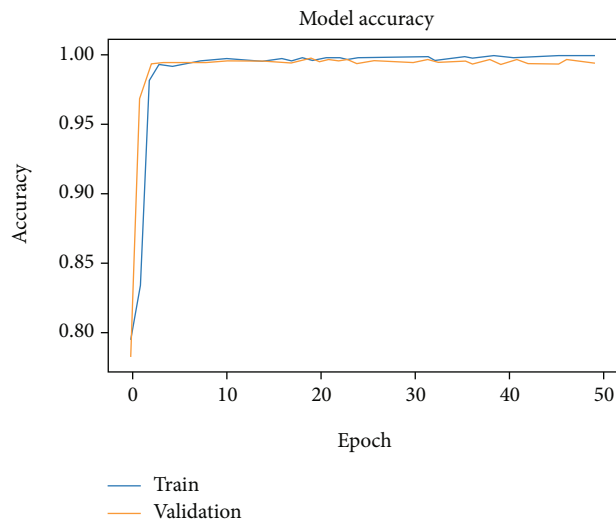
In the mentioned evaluation metrics, TP would be considered the number of true positives, TN is describing the



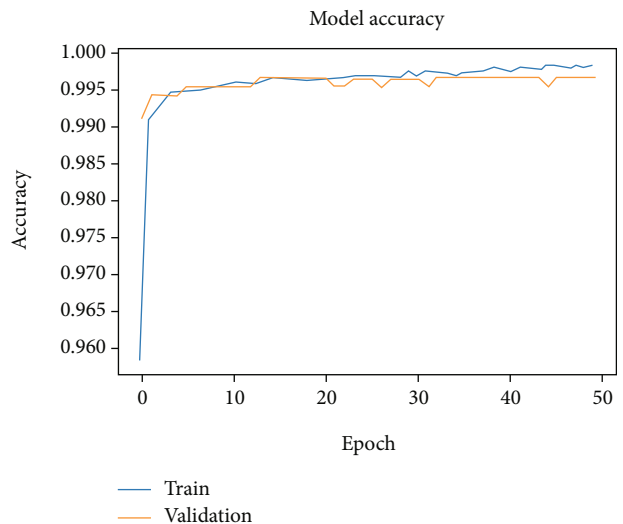
(a) GRU-1



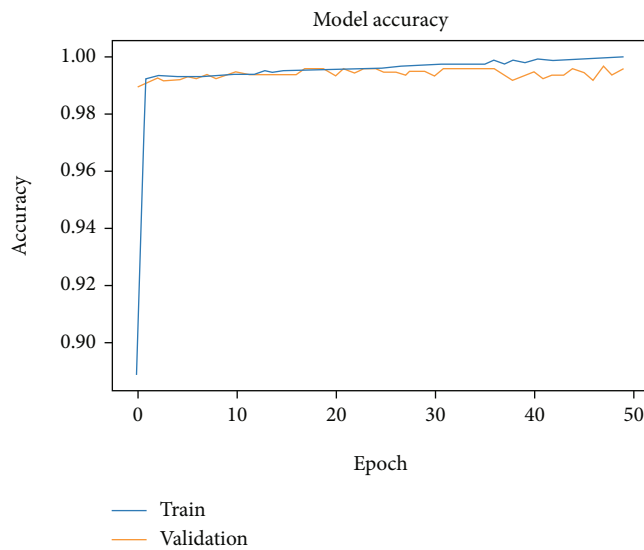
(b) GRU-2



(c) Stacked GRU



(d) LSTM



(e) LSTM-1

FIGURE 9: Continued.

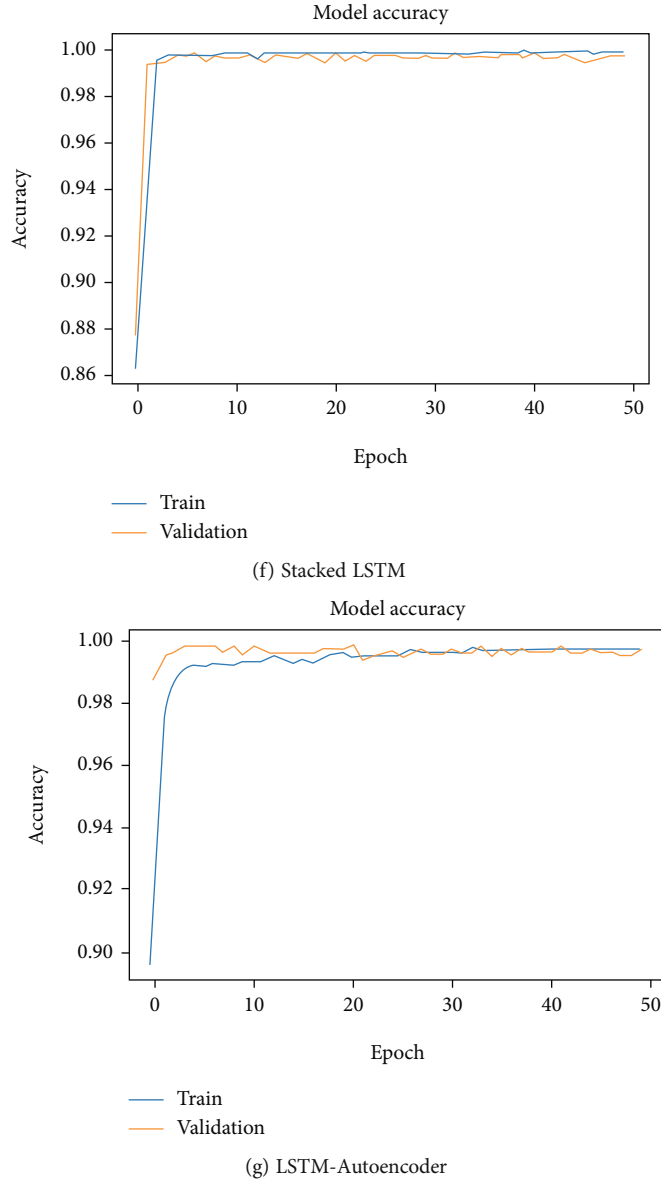


FIGURE 9: Models performance evaluation with accuracy plots.

number of true negatives, *FP* is describing the number of false positives, and *FN* is defined as the number of false negatives. *F1* would be a combination of two metrics as *Precision* and *Recall* and measures the overall performance of the experiments. In such metrics, if both *Precision* and *Recall* are reaching 1, then *F1* would reach 1 as well [39, 65]. Since the dataset is imbalanced, hence, using all evaluation metrics is required to see if the model is well fitted based on *TP*, *TN*, *FN*, and *FP* performances. According to the dataset used in this research, using only one metric like accuracy is not sufficient. Therefore, the *F1* score is a reasonable metric for an evaluation model based on precision and recall. The lower number of *TN* and *FN* could show that the model performs better, and a higher *F1* score assures the well-fitting of the employed model [66]. Therefore, using all the four-evaluation metrics could give a better overview of model performance as presented in the results.

5.12. *Experiments Results and Discussion.* According to performance metrics explained in the previous section, the result of the following experiment has been extracted from different model parameters and configuration which has been presented in the tables as follows for both test and train performance.

5.13. *Experiment Results on Train and Test Data.* Train dataset performance in different model architectures, as well as various activation and loss functions applying to the mentioned dataset, has been mentioned in Table 13.

The result of performance for LSTM-AE in test dataset as well as other model's architectures with varied activation and loss functions applied on the mentioned dataset by the trained model has been mentioned in Table 14 for showing the best-scored model summary in all evaluation metrics.

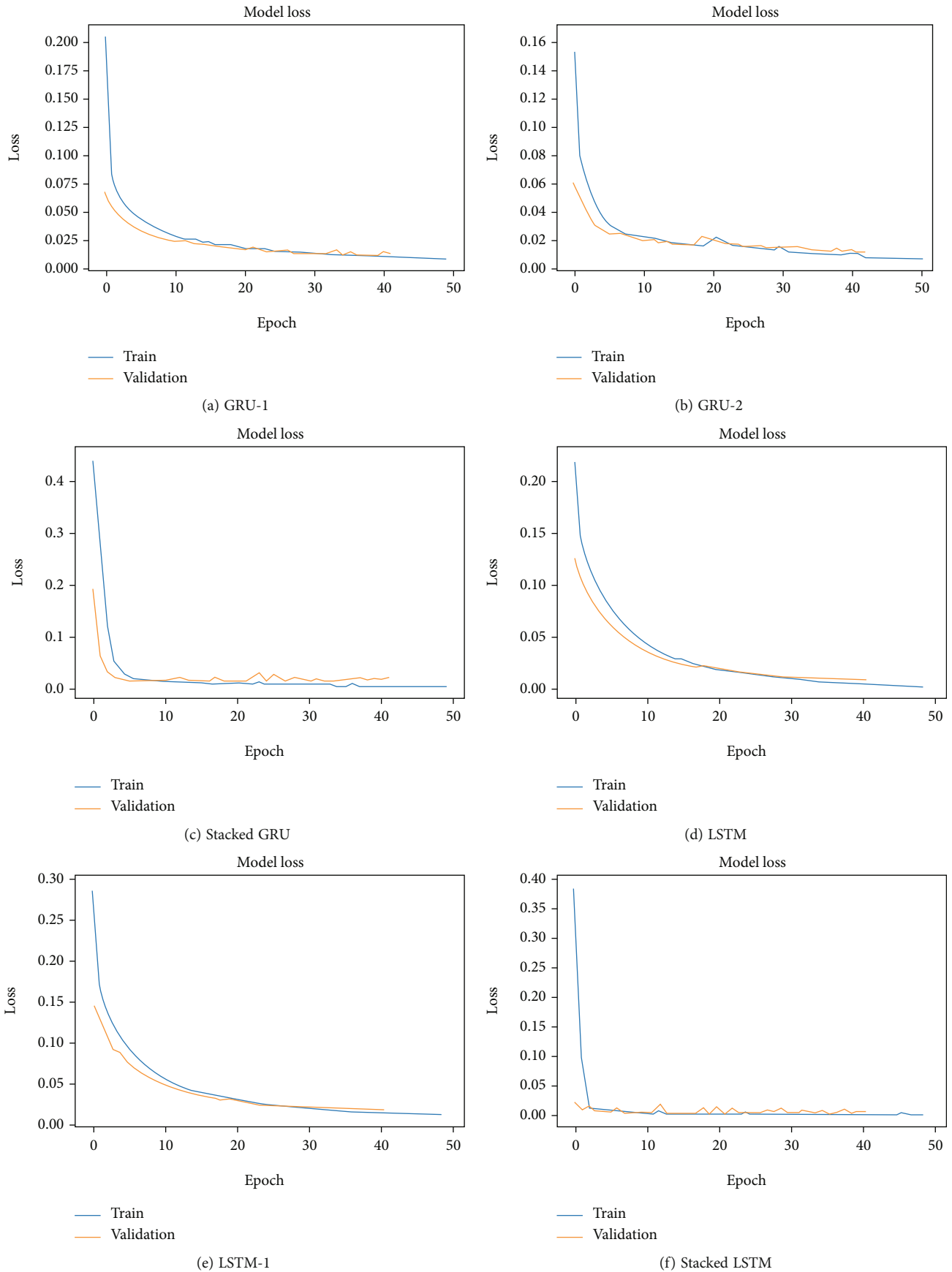


FIGURE 10: Continued.

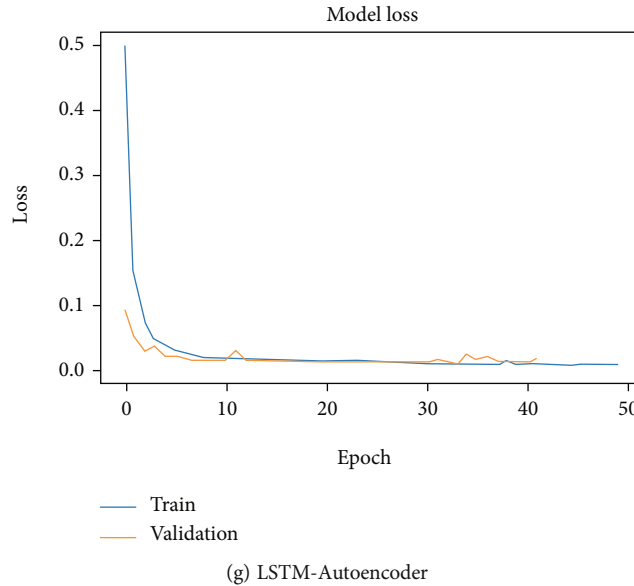


FIGURE 10: Models performance evaluation with loss plots.

5.14. *Model Accuracy.* The following plots (Figures 9(a)–9(g)) show the model accuracy for the models with acceptable performance.

5.15. *Model Loss.* The following plots Figures 10(a)–10(g) show the model loss for the models with proper results to compare the loss output:

6. Discussion

Several experiments were conducted based on different models for parameter optimization, configuration, and architecture revision such as defining the number of units, layers, number of epochs, dropout rate, and the loss and activation function for the classification layer. As mentioned, such a procedure for tuning the model is an empirical process, and the results are the best-achieved score results for each proposed architecture. LSTM-AE, as a proposed model in this paper, showed the best performance among all the models based on RNN architecture. Although stacked networks based on GRU and LSTM have more trainable instances, however, using LSTM-based autoencoder could increase the performance of evaluation metrics with less trainable parameters. For instance, although Stacked LSTM has more parameters (e.g., 102,502 parameters) than all other models, however, it does not show the best performance as adding autoencoder does. This means selecting the right architecture as well as the right parameters could improve the performance. Figure 11 shows the differences between models in terms of trainable parameter numbers based on their architecture.

Additionally, as mentioned in the result section, there are both train and test summary tables. The reason behind presenting each table is to present a comparison between test and train to ensure its ability to classify the detected labels correctly without any overfitting or underfitting issue. As it is clear in the table, different configurations have different

performances in evaluation metrics. For instance, using the mean square error loss function has the least performance in the GRU model, and the sparse categorical entropy has the best performance in all models as well as LSTM-AE which achieved the optimal performance in both train and test which reached 100% for all the evaluation metrics. LSTM-AE performance showed that it could be a proper model to be generalized on unseen data anomaly detection and classification.

Additionally, in the result sections model, accuracy, and model loss have been presented in different plots. Model loss shows a good fit. Since based on the definition of a good fit, training and validation loss shall decrease to a point of stability in the plot with a minimal gap. For instance, final loss values between validation and trained instances should reach near zero. On the other hand, model accuracy plots could be used to see the performance of the model as well as considering whether the model consists of any sort of overfitting or underfitting which in this case the performance for the LSTM-AE showed very well fit. However, GRU-1 and GRU-2 accuracy plots show that these models are not performing as well as LSTM-AE, since in GRU-1 and GRU-2 the validation and train curves are divergent at last.

Computational time and model comparison has been also mentioned in Table 15 which shows the computation time. For such a component with a need for continuous condition monitoring and accurate failure detection, the LSTM-AE computation time would be reasonable due to the high performance of the model, although other models could perform faster. Such computation time is calculated excluding the time for feature extraction time in 50 epochs. The GRU and LSTM-based models with different hyperparameters and configurations have the same computation time and hence categorized as LSTM and GRU in Table 15.

Figure 12 shows an overview of the computational train time.

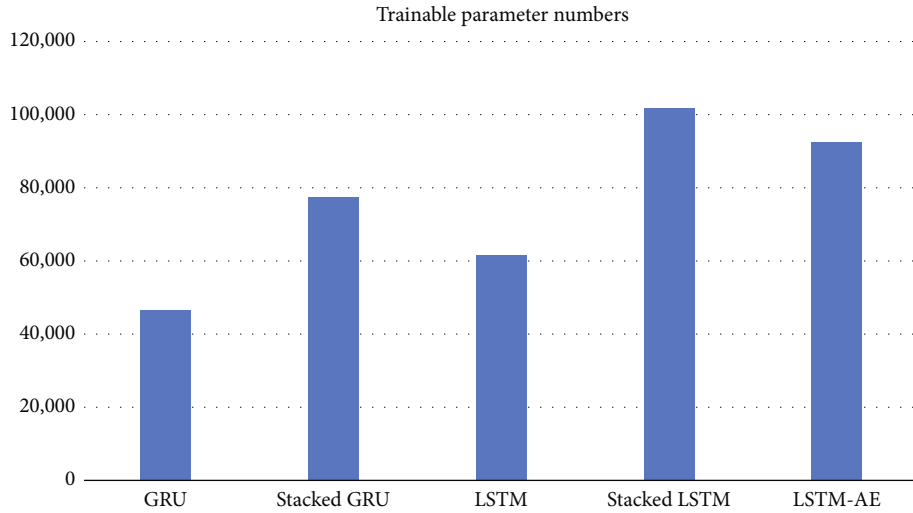


FIGURE 11: Trainable parameters in different models: the figure shows the comparison view for trainable parameters in all the employed deep learning networks based on their architecture (created by authors).

TABLE 15: Computational time comparison among employed models (created by authors).

	Total train time (s)	Avg. epoch time (s) (in 50 epochs)
GRU	52	1.04
LSTM	52	1.04
Stacked LSTM	56	1.12
Stacked GRU	55	1.1
LSTM-AE	801	16.02

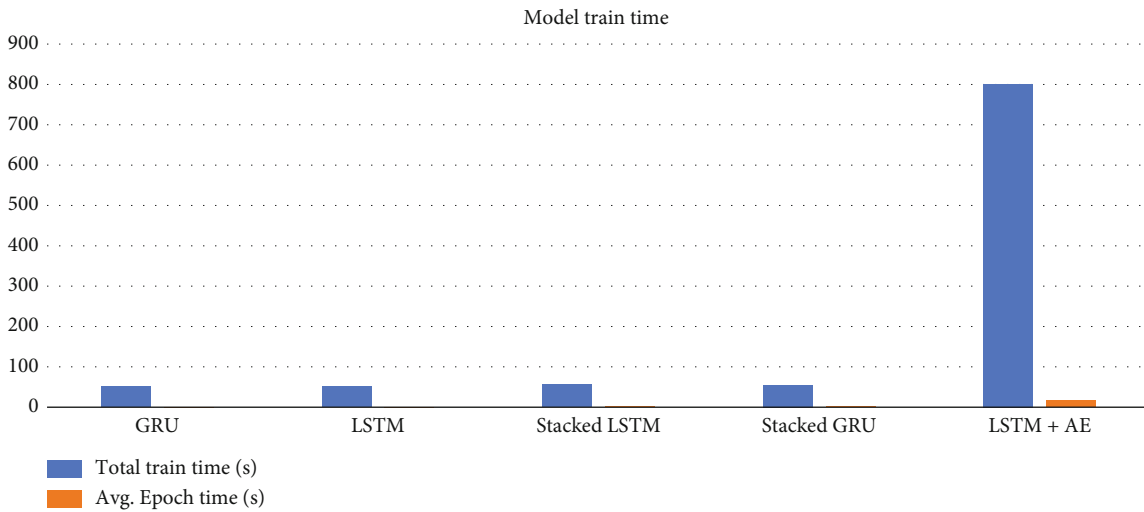


FIGURE 12: Model train time comparison: time comparison of training for all the employed networks which highlights the computations time (created by authors).

Figure 13 could give a better overview which highlights the fact that adding AE to the LSTM architecture has an impact on the improvement of performance in an audio dataset in this paper. This model could be generalized for a larger dataset in time series data as the nature of the AE for handling complex datasets.

There is also a need to highlight the fact that the application of deep learning models is performing better in the larger data set and could give better performance in comparison to other learning algorithms such as statistical methods or machine learning algorithms for conducting a similar task. As shown in Figure 14 [67], the accuracy of the

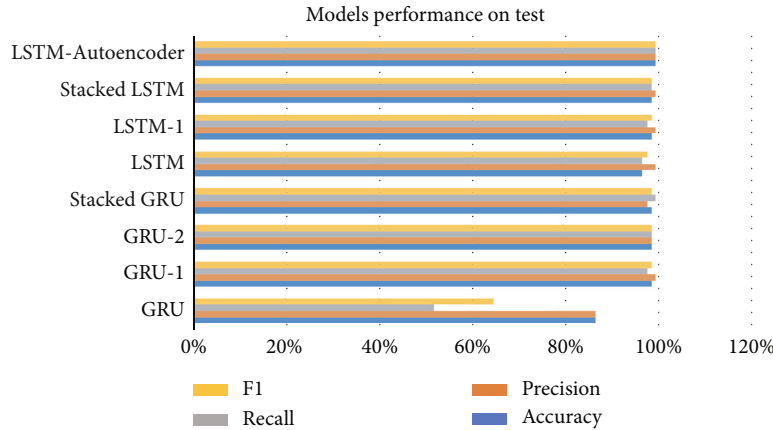


FIGURE 13: Model performance comparison: this figure shows the overview of all the employed models’ performance in the test set (created by authors).

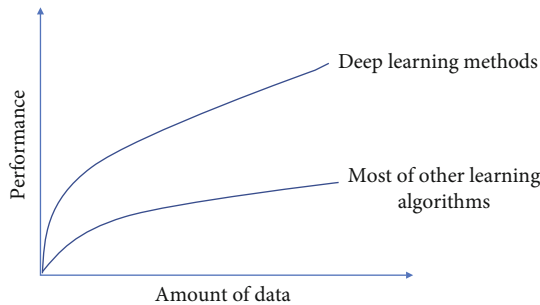


FIGURE 14: Learning curve comparison: the figure shows the performance difference for large dataset between other algorithms and deep learning method [67].

performance of deep learning methods increases in larger datasets which is the reason behind employing deep learning models for the current research challenge for anomaly detection in compressors audio signals [67].

At last, to mention the importance of using deep learning models in the predictive maintenance of such systems, there was a similar research project which has employed an LSTM-based encoder-decoder for anomaly detection. In Pankaj et al., a different number of hidden layers and hyperparameters as well as functions for the purpose of anomaly detection are employed for a time series dataset consisting of the Space shuttle, engine, and ECG multisensor data to consider anomalies based on computing likelihood of anomalies [68]. Additionally, Yu et al. also analyzed different RNN-based variants such as PLSTM and BiLSTM for classification based on SVM and estimated the remained useful life of the machines. This research has employed a random search for hyperparameter tuning and bidirectional RNN autoencoders performed better than unidirectional RNN-based autoencoders [69].

To summarize, in our research, we have employed LSTM-Autoencoders with 8 hidden layers which have been fed by Mel spectrogram and SC features extracted from audio signals from the compressor system Turbo component for the task of anomaly detection and classification with SoftMax function. We have also created a baseline of com-

parison of anomaly detection performance with other RNN variants, and the proposed hybrid model reached a 100% score for all four evaluation metrics. We have also conducted empirical experiments employing different functions and parameters to find the best model configuration for the proposed network.

7. Conclusion

In this research, we have used deep learning neural network models for anomaly detection and classification instead of statistical, machine learning-based, or non-neural network methods which are traditionally used in reliability analysis and predictive maintenance (PdM) analysis and system health and condition monitoring. This paper proposed a hybrid deep learning method for extracting high-level features for feeding the network and a feed-forward neural network in which each cell could remember the important features for the prediction of the output based on LSTM-Autoencoder architecture by minimizing the reconstruction error. The proposed hybrid deep learning-based methods in this research can detect and classify the normal and anomaly audio signals collected from a compressor system effectively. The model has been compared with other baseline neural network architectures based on RNN such as GRU, LSTM, Stacked GRU, and Stacked LSTM. After conducting experiments with different hyperparameters and model configurations, the best performance was based on adding an autoencoder to the LSTM network as a hybrid model. Such a method could be proposed for increasing the reliability of the compressor systems with the ability to detect defects in time.

In the proposed model Mel Spectrogram and SC have been used as features for feeding the network for training. The combination of multiple-resource features and hybrid models showed a 100% score in all four-evaluation metrics such as accuracy, precision, recall, and F1 in the LSTM-based autoencoder in both test and train results as the best-scored model.

This paper could also highlight the fact that employing the deep learning method in predictive maintenance tasks could maintain the safety of technicians and staff by avoiding any dangerous risk of failures for the compressor system components that could harm them. Additionally, it could reduce the risk of failures of the compressor's system which could increase system reliability at last by in-time anomaly detection.

8. Future Work

Using more balanced datasets as well as increasing dataset size could be a baseline for future research. Additional to that using autoencoder along with LSTM enables the capability to define reconstruction error which could be used as a baseline for defining a threshold for anomaly detectors to generalize and detect unseen anomalies. Alongside applying AE, variational autoencoder could be also used for anomaly detection combined with RNN-based networks. VAE in comparison to AE is working with distribution and reconstructing the probabilities from input data which could be used as a baseline for anomaly detection in low probabilities.

Data Availability

The auditory data used to support the findings of this study have not been made available because of third-party rights on commercial confidentiality.

Disclosure

This research project was performed as part of the employment of the authors as graduate student research assistants and faculty members of Lamar University.

Conflicts of Interest

The research authors declare that they have no conflict of interest about this work.

Acknowledgments

This research is conducted in the Departments of Industrial and Systems Engineering and Computer Science, Lamar University, and dataset provided by Well Checked Systems.

References

- [1] B. Guo, W. C. Lyons, and A. Ghalambor, *Petroleum Production Engineering: A Computer-Assisted Approach*, Gulf Professional Pub, 2007.
- [2] T. Abbasi, K. H. Lim, N. S. Rosli, I. Ismail, and R. Ibrahim, "Development of predictive maintenance interface using multiple linear regression," in *International Conference on Intelligent and Advanced System (ICIAS)*, pp. 1–5, Kuala Lumpur, Malaysia, 2018.
- [3] F. Jahandideh, S. N. Raman, M. Jamil, and Z. I. Syed, "Carbon footprint assessment in the life-cycle design of concrete structures in the tropics: a case study of residential buildings in Malaysia," *Journal of Design and Built Environment*, vol. 20, no. 2, pp. 27–34, 2020.
- [4] O. Motaghare, A. S. Pillai, and K. I. Ramachandran, "Predictive maintenance architecture," in *IEEE International Conference on Computational Intelligence and Computing Research (ICCC)*, pp. 1–4, Madurai, India, 2018.
- [5] F. Jahandideh, S. N. Raman, M. Jamil, and R. Prakash, "Life-cycle sustainability assessment in the design of concrete structures in the tropics: a fundamental analysis," *Journal of Design and Built Environment*, vol. 20, no. 3, pp. 43–51, 2020.
- [6] F. Hua, X. Peng, and X. Ruoyan, "KLS-A: a full-life-time anomaly detection method," in *International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp. 489–493, Beijing, China, 2020.
- [7] M. Li and Z. Wang, "Deep learning for high-dimensional reliability analysis," *Mechanical Systems and Signal Processing*, vol. 139, article 106399, 2020.
- [8] H. S. Nam, Y. K. Jeong, and J. W. Park, "An anomaly detection scheme based on LSTM autoencoder for energy management," in *International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1445–1447, Jeju, Korea, 2020.
- [9] X. Li, W. Zhangc, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliability Engineering and System Safety*, vol. 182, pp. 208–218, 2019.
- [10] M. Jafari Kang, S. Zohoori, E. Abbasi, Y. Li, and M. Hamidi, *Predicting the price of second-hand vehicles using data mining techniques*, University Library of Munich, Germany, 2019.
- [11] J. Ma, "Machine learning in predicting diabetes in the early stage," in *2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pp. 167–172, Taiyuan, China, 2020.
- [12] H. Merdassi, W. Barhoumi, and E. Zagrouba, "Semi-supervised image classification in large datasets by using random forest and fuzzy quantification of the salient object," in *International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM)*, pp. 1–5, Paris, France, 2014.
- [13] Z. F. Yan, Y. L. Shen, W. J. Liu, J. M. Long, and Q. Wei, "An E-commerce coupon target population positioning model based on random forest and extreme gradient boosting," in *2018 11th International Congress on Image and Signal Processing, Bio-Medical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, Beijing, China, 2018.
- [14] M. Munir, M. A. Chattha, A. Dengel, and S. Ahmed, "A comparative analysis of traditional and deep learning-based anomaly detection methods for streaming data," in *18th IEEE International Conference On Machine Learning and Applications (ICMLA)*, pp. 561–566, Boca Raton, FL, USA, 2019.
- [15] M. Farahani, 'Anomaly detection on gas turbine time-series' data using deep LSTM-autoencoder, Ph.D. thesis, Umeå University, Faculty of Science and Technology, Department of Computing Science, 2021.
- [16] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): a fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track*, vol. 9, 2012.
- [17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39, 2012.
- [18] Y. Zhao and M. K. Hryniewicki, "XGBOD: improving supervised outlier detection with unsupervised representation learning," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Rio de Janeiro, Brazil, 2018.

- [19] Z. Jiulong, G. Luming, Y. Su, S. Xudong, and L. Xiaoshan, "Detecting Chinese calligraphy style consistency by deep learning and one-class SVM," in *2nd International Conference on Image, Vision and Computing (ICIVC)*, pp. 83–86, Chengdu, China, 2017.
- [20] E. R. Setyaningsih and I. Listiowarni, "Categorization of exam questions based on bloom taxonomy using Naïve Bayes and Laplace smoothing," in *3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pp. 330–333, Surabaya, Indonesia, 2021.
- [21] Q. Zhou, W. Lan, Y. Zhou, and G. Mo, "Effectiveness evaluation of anti-bird devices based on random forest algorithm," in *7th International Conference on Information, Cybernetics, and Computational Social Systems, ICCSS 2020*, pp. 743–748, Guangzhou, China, 2020.
- [22] T. E. Tremain, "The government standard linear predictive coding algorithm: LPC-10," *Speech Technology*, vol. 1, no. 2, pp. 40–49, 1982.
- [23] J. Mrva, Š. Neupauer, L. Hudec, J. Ševcech, and P. Kapec, "Decision support in medical data using 3D decision tree visualisation," in *E-Health and Bioengineering Conference (EHB)*, pp. 1–4, Iasi, Romania, 2019.
- [24] E. Wong, *Media Review*, InnovAiT, 2021.
- [25] R. Deo, P. Samui, and S. S. Roy, *Predictive Modelling for Energy Management and Power Systems Engineering*, Elsevier, 2021.
- [26] L. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Pearson Education India, 1978.
- [27] S. Kavitha, S. Varuna, and R. Ramya, "A comparative analysis on linear regression and support vector regression," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, pp. 1–5, Coimbatore, 2016.
- [28] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: a deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2018.
- [29] P. Mondal, L. Shit, and S. Goswami, "Study of effectiveness of time series modeling (Arima) in forecasting stock prices," *International Journal of Computer Science, Engineering and Applications*, vol. 4, no. 2, pp. 13–29, 2014.
- [30] P. Mobtahej, M. Hamidi, X. Zhang, and J. Zhang, "Deep learning-based anomaly detection for compressors using audio data," in *Annual Reliability and Maintainability Symposium, (RAMS)*, pp. 1–7, Orlando, FL, USA, 2021.
- [31] S. Wang, J. Feng, and G. Liu, "Application of seasonal time series model in the precipitation forecast," *Mathematical and Computer Modelling*, vol. 58, no. 3–4, pp. 677–683, 2013.
- [32] X. Zhang, *Anomaly Compressor Audio Data Detection and Classification using Deep Learning Techniques*, [M.S. thesis], Lamar University - Beaumont, 2021.
- [33] T. P. Khanh and K. Nguyen, "A new dynamic predictive maintenance framework using deep learning for failure prognostics," *Reliability Engineering & System Safety*, vol. 188, pp. 251–262, 2019.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.
- [35] X. Tian and C. Chen, "Modulation pattern recognition based on Resnet50 neural network," in *2nd International Conference on Information Communication and Signal Processing (ICICSP)*, pp. 34–38, Weihai, China, 2019.
- [36] Z. Fang, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.
- [37] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *International Conference on Web-Age Information Management*, Macau, China, 2014.
- [38] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," *CUIDADO IST Project Report*, vol. 54, pp. 1–25, 2004.
- [39] O. I. Provotar, Y. M. Linder, and M. M. Veres, "Unsupervised anomaly detection in time series using lstm-based autoencoders," in *In IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, pp. 513–517, Kyiv, Ukraine, 2019.
- [40] H. Gunasekaran, K. Ramalakshmi, A. R. M. Arokiaj, S. D. Kanmani, C. Venkatesan, and C. S. G. Dhas, "Analysis of DNA sequence classification using CNN and hybrid models," *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 1835056, 12 pages, 2021.
- [41] Z. Tan and P. Pan, "Network fault prediction based on CNN-LSTM hybrid neural network," in *International Conference on Communications, Information System and Computer Engineering (CISCE)*, pp. 486–490, Haikou, China, 2019.
- [42] A. Sahar and H. Dongsoo, "An LSTM-based indoor positioning method using Wi-Fi signals," in *ACM International Conference Proceeding Series*, pp. 1–5, Vegas, USA, 2018.
- [43] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*, Springer International Publishing, Cham, 2017.
- [44] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM model for short-term individual household load forecasting," *IEEE Access*, vol. 8, pp. 180544–180557, 2020.
- [45] K. Adem and S. Kilicarslan, "Performance analysis of optimization algorithms on stacked autoencoder," in *3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2019 - Proceedings*, pp. 1–4, Ankara, Turkey, 2019.
- [46] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '20)*, pp. 37–45, New York, NY, USA, 2020.
- [47] D. Goularas and S. Kamis, "Evaluation of deep learning techniques in sentiment analysis from Twitter data," in *International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pp. 12–17, Istanbul, Turkey, 2019.
- [48] T. Giannakopoulos and A. Pikrakis, *Introduction to Audio Analysis: A MATLAB Approach*, Academic Press, Inc., USA, 1st edition, 2014.
- [49] M. H. Tanveer, M. Sabbagh, A. Thomas, B. M. Imran, and M. H. Soomro, "RNN and Mel-spectrogram based acoustic balancing and bouncing of ping pong ball," in *International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, pp. 16–20, Tirana, Albania, 2020.
- [50] M. H. Tanveer, H. Zhu, W. Ahmed, A. Thomas, B. M. Imran, and M. Salman, "Mel-spectrogram and deep CNN based representation learning from bio-sonar implementation on UAVs," in *International Conference on Computer, Control and Robotics (ICCCR)*, pp. 220–224, Shanghai, China, 2021.

- [51] K. Molugaram and G. Shanker Rao, "Analysis of time series," in *Statistical Techniques for Transportation Engineering*, pp. 463–489, Butterworth-Heinemann, 2017.
- [52] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to Time Series and Forecasting*, Springer, New York, NY, USA, 2002.
- [53] K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar, M. Kawaji, and T. M. Brenza, "Forecasting of COVID-19 using deep layer recurrent neural networks (RNNs) with gated recurrent units (GRUs) and long short-term memory (LSTM) cells," *Chaos, Solitons & Fractals*, vol. 146, article 110861, 2021.
- [54] M. Pavithra, K. Saruladha, and K. Sathyabama, "GRU based deep learning model for prognosis prediction of disease progression," in *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC*, pp. 840–844, Erode, India, 2019.
- [55] T. Hu and C. Meinel, "An investigation of fine-tuning pre-trained model for MR-to-text generation," in *19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1006–1009, Miami, FL, USA, 2020.
- [56] J. G. Liu, X. J. Yang, Y. Y. Feng, and P. Cui, "New fractional derivative with sigmoid function as the kernel and its models," *Journal of Physics*, vol. 68, pp. 533–541, 2020.
- [57] V. Kotu and B. Deshpande, "Chapter 10- deep learning," in *Data Science*, V. Kotu and B. Deshpande, Eds., pp. 307–342, Morgan Kaufmann, 2019.
- [58] A. F. T. Martins and R. F. Astudillo, *From Softmax to Sparse-max: a Sparse Model of Attention and Multi-Label Classification*, ICML, 2016.
- [59] T. Varun, J. Mohana, R. Durganath, and M. I. Hussan, "Prediction of animal vocal emotions using convolutional neural network," *International Journal of Scientific & Technology Research*, vol. 9, pp. 6007–6011, 2017.
- [60] C. H. Chen, P. H. Lin, J. G. Hsieh, S. L. Cheng, and J. H. Jeng, "Robust multi-class classification using linearly scored categorical cross-entropy," in *3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, pp. 200–203, Kaohsiung, Taiwan, 2020.
- [61] Y. Sai, R. Jinxia, and L. Zhongxia, "Learning of neural networks based on weighted mean squares error function," in *second International Symposium on Computational Intelligence and Design*, pp. 241–244, Changsha, China, 2009.
- [62] Z. Que, Y. Liu, C. Guo, X. Niu, Y. Zhu, and W. Luk, "Real-time anomaly detection for flight testing using autoencoder and LSTM," in *International Conference on Field-Programmable Technology (ICFPT)*, pp. 379–382, Tianjin, China, 2019.
- [63] S. Maleki, S. Maleki, and N. R. Jennings, "Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering," *Applied Soft Computing*, vol. 108, article 107443, 2021.
- [64] C. Zhiyong, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, <https://arxiv.org/abs/1801.02143>.
- [65] N. DeCastro-García, A. L. M. Castañeda, and M. Fernández-Rodríguez, "Machine learning for automatic assignment of the severity of cybersecurity events," *Computational and Mathematical Methods*, vol. 2, no. 1, 2020.
- [66] S. Naddaf-Sh, M.-M. Naddaf-Sh, A. R. Kashani, and H. Zargarzadeh, "An efficient and scalable deep learning approach for road damage detection," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 5602–5608, Atlanta, GA, USA, 2020.
- [67] M. Kraus, S. Feuerriegel, and A. Oztekin, "Deep learning in business analytics and operations research: models, applications and managerial implications," *European Journal of Operational Research*, vol. 281, no. 3, pp. 628–641, 2020.
- [68] M. Pankaj, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. M. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, <https://arxiv.org/abs/1607.00148>.
- [69] W. Yu, I. Y. Kim, and C. Mechefske, "Analysis of different RNN autoencoder variants for time series classification and machine prognostics," *Mechanical Systems and Signal Processing*, vol. 149, article 107322, 2021.