Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 1637 – 1653

doi:10.55730/1300-0632.3870

Anomaly detection in rotating machinery using autoencoders based on bidirectional LSTM and GRU neural networks

Krishna Chandra Patra[1,*], **Rabinarayan Sethi**[2], **Dhiren Kumar Behera**[2]
[1]Biju Patnaik University of Technology, Rourkela, India
[2]Indira Gandhi Institute Of Technology, Sarang, India

Received: 21.11.2021 • **Accepted/Published Online:** 22.03.2022 • **Final Version:** 31.05.2022

Abstract: A time series anomaly is a form of anomalous subsequence that indicates future faults will occur. The development of novel techniques for detecting this type of anomaly is significant for real-time system monitoring. Several algorithms have been used to classify anomalies successfully. However, the time series anomaly detection algorithm was not studied well. We use a new bidirectional LSTM and GRU neural networks-based hybrid autoencoder to detect if a machine is operating normally in this research. An autoencoder is trained on a set of 12 features taken from healthy operating data gathered promptly after a planned maintenance period using vibration sensors. The features taken from new data are then reconstructed using the trained model. If the model accurately reconstructs the features, the machine is in good working order. If the reconstruction exceeds a certain error threshold, the machine is functioning strangely and needs to be serviced.

Key words: Autoencoders, anomaly detection, deep learning, predictive maintenance

1. Introduction

Anomaly detection in time series is a critical subject in the fields of computer science and data mining [1–4]. It was widely used in a variety of practical uses, including signal processing, pattern identification [5], mathematical finance, forecasting the weather [6], and control engineering [7]. It has become a requirement in the industrial sector, as faults that go undetected can result in a catastrophic tragedy [8]. Industrial systems are subjected to a great deal of stress daily and are failure-prone, thus detection of anomalies can help to improve system availability and performance. This has a direct impact on productivity and lowers operating and maintenance expenses [9]. As a result, several studies in this field may be found in a variety of industries, including automotive [10], manufacturing [11], energy [12], Sensing networks in the industry [13], or even medicine, which uses a variety of pictures [14].

A time-series anomaly is described as an out of the ordinary pattern that deviates from predicted behaviour. There have been three types of categories of time series anomalies: a single point, a context, and a group [3]. Individual samples that exceed the usual range are referred to as point anomalies and can be discovered using an ultralimit alert. When the sequences' context shifts, contextual anomalies develop, which is always tightly tied to the temporal element. Multiple occurrences of data may constitute an anomaly as a sequential sequence, however, the individuals in this series may be irrelevant. As a result, it is difficult for models in detecting anomalies to efficiently capture distinct properties of several abnormalities in a time series.

*Correspondence: kcpmechcvrce@gmail.com

As a result of the growth of industry and the Internet of Things (IoT), in recent years, there have been breakthroughs in spotting anomalies in time series data [15]. Multiple sensors are equipped to capture a large number of time series, and technology has provided more efficient services to companies and dependable monitoring systems. Multiple events govern industrial systems, making it difficult to identify characteristics and particular locations in anomalous time series with several variables. As a result, intricate industrial systems include diverse sensors, which can have a variety of properties, sizes, and characteristics, as well as high dimensionality and dependence on space and time [16]. This fact frequently means that data has been cleaned, valuable features extracted, or the dimensionality of the data has been reduced [17]. This is usually a time-consuming procedure that necessitates domain knowledge. Aside from these difficulties, there are other intrinsic problems, like the difficulty to design boundaries between normal and abnormal data, and there is a lot of noise caused by a faulty sensor or incorrect measurements, which can cause false alarms to occur. Due to the scarcity of anomalous observations, data imbalance [18] is a prevalent problem in anomaly detection settings, affecting the models' robustness in detecting anomalies.

Many researchers have looked into time series modelling, particularly with traditional statistical approaches like ARIMA [19], HMM [20], and GMM [21]. SVM [22] and Random Forest [23] for example, is a type of machine learning method. Statistical methods, on the other hand, have a hard time dealing with unknown statistical traits and data with a lot of dimensions, whereas methods for machine learning necessitate the generation and pre-processing of time-consuming features. Fortunately, for multitime series modelling, deep learning has shown promise, and numerous studies have used deep learning methods to solve these difficulties in the identification of anomalies in multiple time series. [8] provides a CNN and RNN-based supervised anomaly detection approach, which shows promise on industrial multitime series. To protect information technology infrastructures from harmful malware attacks, unique statistical analysis and autoencoder are used to create an intelligent method that has been devised [24]. Hu [25] uses Faster R-CNN to distinguish items in the scene and trains an enhanced detection using an SVM-based classifiers and locate anomalous behaviours. To anticipate anomalies in crowd frame sequences, a suggested method [26] leverages a collection of fine-tuned CNN architectures for training variations of SVM classifiers. To answer the industrial anomaly detection problem, Fan [27] introduces a unique anomaly detection using a hybrid unsupervised approach that combines convolutional neural network autoencoder with Gaussian process regression.

Recurrent neural networks, or LSTM networks, have demonstrated anomaly detection and performance in sequence learning problems that is at the cutting edge [28, 29]. The input pattern is converted into a fixed-length latent vector representation by a long short-term memory encoder. The decoder, which is another LSTM network, then reconstructs the input sequence using the latent representation. When compared to autoencoders, autoencoders based on LSTM achieve even greater results in anomaly detection [30]. However, only complicated LSTM networks with significant computational complexity and enormous memory requirements may produce considerable gains in detection accuracy [31]. Intrusion detection using a deep neural network (CNN + Bi-LSTM) has been suggested by Jiang et al. [32]. Neural networks with gated recurrent units (GRUs) have been discovered to have simpler architectures and training times faster than LSTM neural networks. Yuan and Tian [33] developed a GRU neural network-based dynamic process fault detection technique, to extract dynamic properties and categorise industrial processes, GRU neural networks were used. Afrasiabi et al. [34] used a differential protection strategy that integrated light-gated recurrent neural networks units and CNNs to identify inrush current in power transformers caused by an internal fault. However, despite their many advantages, these data-driven approaches necessitate a significant number of abnormal samples, which can be difficult to get in

reality. In addition, the detection of aberrant operating conditions in limited samples still has a lot of room for improvement.

Unsupervised machine intrusion detection system in industry time-series data employing a novel framework, BiLSTM-GRU, is proposed in this research, and several enhancements are given to address those critical issues. The following are the primary contributions of this work:

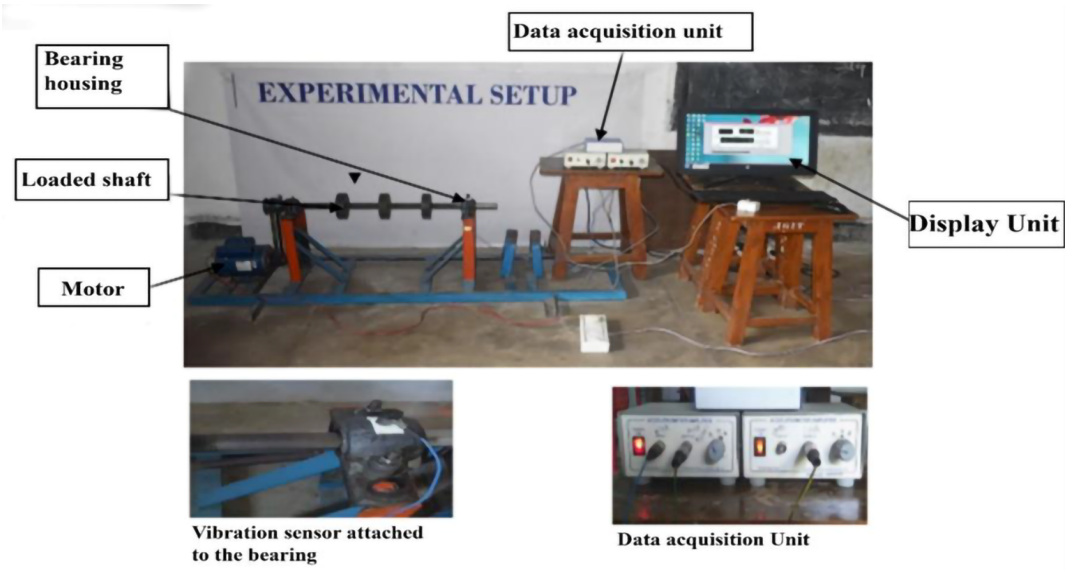
- While a model can be trained on raw data, it is often preferable to extract key features for the best accuracy. After extracting key features from industrial time-series data, we use the t-test, Wilcoxon, and Bhattacharyya tests to rank them.
- Second, BiLSTM-GRU is offered as a unique framework for anomaly detection in industrial time series. The use of a sliding window to iteratively update the algorithm's parameters, which allows for quick identification of abnormalities, necessitates a constant amount of data, ensuring linear complexity in space and time.
- In machine anomaly detection, this model can be used to perform regression and classification tasks. This model will assist to increase model resilience in time series anomaly detection when normal and abnormal data are imbalanced.
- Lastly, tests on data sets are carried out to verify the efficacy of the proposed frameworks and effectiveness of a unique threshold-setting technique, with the outcomes demonstrating that our strategy outperforms comparable models in terms of robustness and performance in detecting anomalies under various imbalanced dataset ratios.
- The model combines the memory strength of a linear regression model with the generalisation strength of a deep GRU neural network model for increased accuracy.

2. Dataset preprocessing

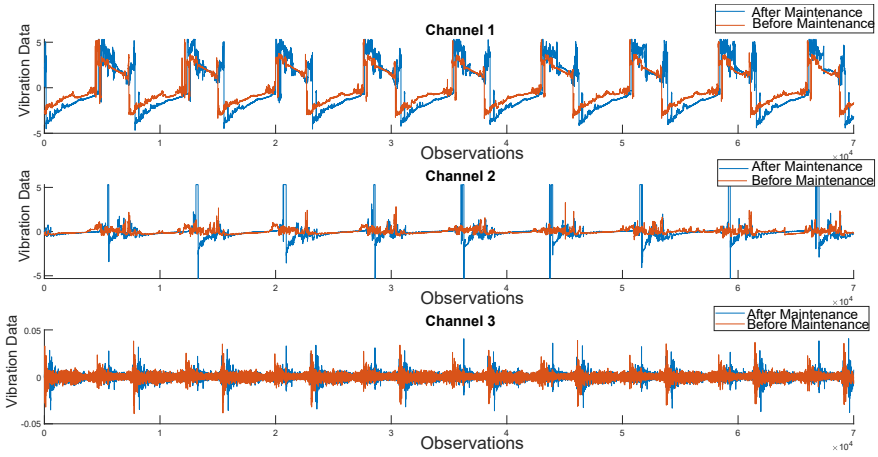
To create the dataset, the test rig Figure 1a consists of a 0.5 hp electric motor attached to the left, driving a shaft on which two bearings are mounted at both ends. The vibration data was obtained using a 3-axis accelerometer attached to the housing magnetically through Ch1, Ch2, and Ch3. The data was collected with a sampling rate of 70,000 samples per second from our test rig before and after scheduled maintenance periods and was processed using MATLAB. We can presume that all of the "after" data is healthy because maintenance was completed appropriately. The "before" data, on the other hand, is less clear: the machine may have been brought down for maintenance owing to a fault, but machines are frequently taken offline for scheduled maintenance even when they are working normally. Let's view the data before and after maintenance to better comprehend it. We can observe that the data before and after maintenance Figure 1b looks notably different if we go through each of the members.

3. Featurization

Feature engineering is critical in every process of machine learning and can have a substantial impact on an algorithm's performance. The machine learning algorithm can benefit from feature engineering in identifying the underlying patterns and so increase the model's accuracy. Featurization of raw signal data requires extracting features from raw signals in a variety of domains, including time, time-frequency, and so on. Signals of vibration from machinery components are regarded as nonstationary in general. The term "nonstationary signals" refers



(a) Experimental test rig.



(b) Visualize data before and after maintenance.

Figure 1. Experimental test rig and data visualization before and after maintenance.

to transmissions whose frequency change over time [35]. To convey the changing nature of the character over time present in a signal, it is necessary to extract time-domain and time-frequency-domain features. Multiple temporal and time-frequency domain features were recovered from raw signal data in this paper. Mean, variance, standard deviation, and root mean square (RMS) are some of the statistical temporal domain traits we glean. Since these signals are not stationary, features like kurtosis and skewness are retrieved as well. For anomaly, the kurtosis increases its value, and skewness shifts to the negative or positive side. Dimensionless characteristics such as the crest factor, shape factor, and impulse factor are also derived in addition to previous statistical features. The shape factor is influenced by its shape, although it is unaffected by its dimension. Signal to noise ratio (SNR) is the ratio of fundamental signal amplitude to noise signal amplitude. Total harmonic distortion (THD) is the ratio of the sum of the fundamental signal component's harmonics. For characterisation, the first 5 to 6 harmonic components are usually used. The signal to noise and distortion ratio (SINAD) is the

proportion of the signal amplitude (in rms) to the sum of the other spectral components (in rms). Its value is approximately equal to THD + noise. Table 1 lists all 10 features retrieved from the raw signal data, as well as the mathematical techniques utilised to extract them.

Table 1. The mathematical procedures were utilised to calculate the values of the features.

Feature	Formula
Mean	$\text{Mean } = \frac{1}{n}\sum_{i=1}^{n} x_i$
Peak value	$\text{PeakValue } = max\left(\lvert x_i \rvert\right)$
Variance	$\text{Var } = \frac{1}{n}\sum_{i=1}^{n} \left(x_i - x\right)^2$
Root mean square	$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$
Impulse factor	$IF = \frac{PeakValue}{\frac{1}{n}\sum_{i=1}^{n} \lvert x_i \rvert}$
Clearance factor	$\text{ClF} = \frac{Peak\text{Value}}{\left(\frac{1}{N}\sum_{i=1}^{N} \sqrt{\lvert x_i \rvert}\right)^2}$
Shape factor	$SF = \frac{RMS}{\frac{1}{n}\sum_{i=1}^{n} \lvert x_i \rvert}$
Crest factor	$CF = \frac{PeakValue}{RMS}$
Kurtosis	$\text{Kurt } = \frac{\sum_{i=1}^{n} \left(x_i - x\right)^4}{n \times var^2} - 3$
Skewness	$\text{Skw } = \frac{\frac{1}{n}\sum_{i=1}^{n} \left(x_i - \overline{x}\right)^3}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(x_i - x\right)^2}^3}$

While a model can be trained on raw data, it is often preferable to extract key features for the best accuracy. Review and preprocess our data interactively using MATLAB software, establish data label as a condition variable, then extract time and frequency-domain features and rank them to see which are the most successful. This is because the data label indicates the machine's condition: before and after maintenance, and we can study these datasets individually. The goal is to teach a model to distinguish between these two scenarios. The distributions divided by labels for various features retrieved from Ch1, Ch2, and Ch3 are visualized in Figure 2.

When a 3-axis accelerometer is employed, one true radial measurement, one tangential measurement, and one axial measurement are obtained, depending on the position of the accelerometer. As a result, each channel is equally important. We need to figure out which features are optimal for each channel. The histogram plot and feature importance ranking approach are used to find the best features. The figure 2 shows the parameters that determine the histogram's content and resolution. The resulting feature differs in Ch1, Ch2, and Ch3 because the vibration amplitude varies in the X, Y, and Z axes. By evaluating which features clearly separate blue data from orange data, we may get a general notion of which ones are effective. THD, SINAD, and SNR appear to be effective for channel-3, with relatively minor overlap. SINAD and mean, on the other hand, have a lot of overlap for channel-1. As a result, these characteristics appear to be ineffective. In addition, as described in subsection 3.1, we apply feature importance ranking to improve feature selection.

3.1. Feature importance ranking for deep learning

The task of measuring the contributions of individual input features (variables) to the performance of an unsupervised learning model is known as feature importance ranking (FIR) in deep learning. FIR has become one of the most powerful instruments in explainable/interpretable AI [36] for facilitating the understanding of a

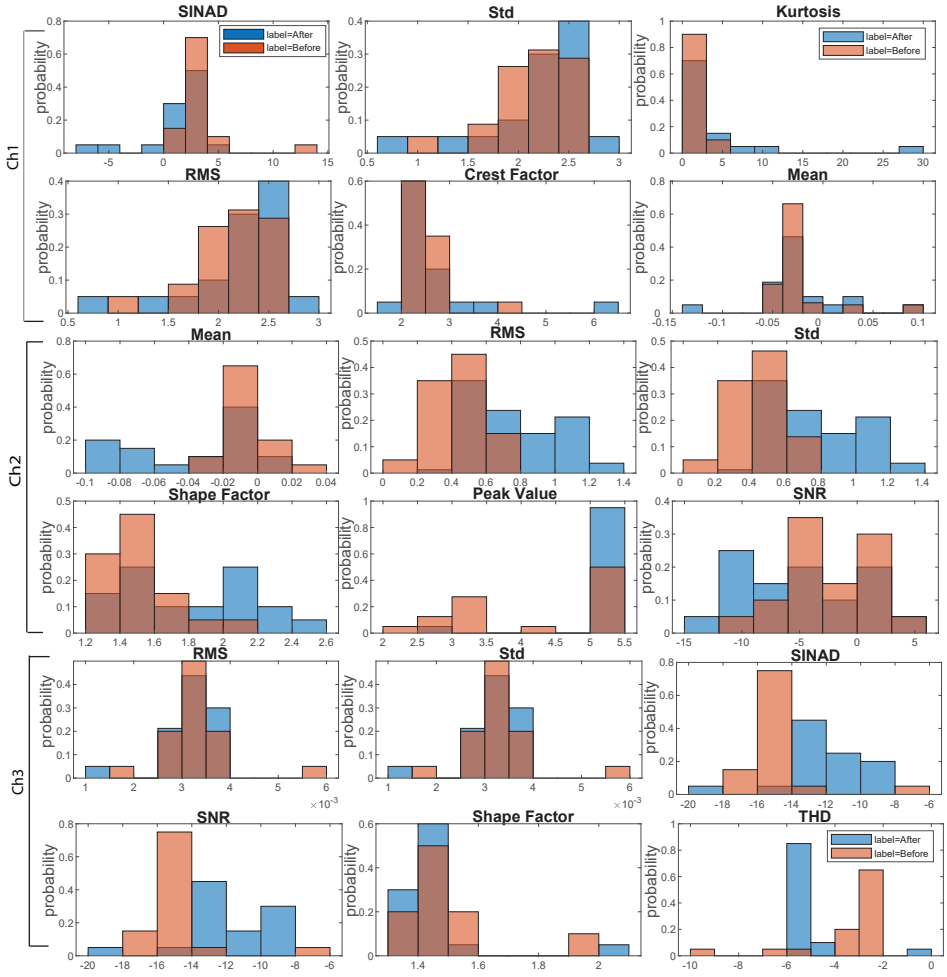


Figure 2. Histograms of features retrieved from the raw signal data.

learning system's decision-making and the discovery of essential features in a certain domain. To rank all of the features extracted from the raw signal data, we employ the t-test, Wilcoxon rank-sum test, and Bhattacharyya distance.

A t-test is any statistical hypothesis test in which the test statistic follows a Student's t-distribution when the null hypothesis is true. The Wilcoxon rank-sum test is a nonparametric test of the null hypothesis that the likelihood of X being larger than Y given randomly picked values X and Y from two populations is equal to the probability of Y being greater than X. For two classes χ_1, χ_2 the Bayes classifier's minimal attainable classification error is expressed as:

$$P_e = \int_{-\infty}^{\infty} min\left[P\left(\chi_i\right) p\left(x \mid \chi_i\right), P\left(\chi_j\right) p\left(x \mid \chi_j\right)\right] dx. \tag{1}$$

In the general case, analytic computation of Equation (1) is not possible. An upper bound, on the other hand, can be calculated. The inequality serves as the foundation for the derivation.

$$min\left[a, b\right] \leq a^s b^{1-s} \text{ for } a, b \geq 0, \text{ and } 0 \leq s \leq 1 \tag{2}$$

Combining Equation (1) and Equation (2), we get

$$P_e \leq P(\chi_i)^s P(\chi_j)^{1-s} \int_{-\infty}^{\infty} p(x \mid \chi_i)^s p(x \mid \chi_j)^{1-s} dx \equiv {}_{CB} \tag{3}$$

${}_{CB}$ is recognized as the Chernoff bound. By decreasing ${}_{CB}$ concerning s, the minimum bound can be found. The bound results for $s = {}^1/_2$ in a specific form:

$$P_e \leq {}_{CB} = \sqrt{P(\chi_i) P(\chi_j)} \int_{-\infty}^{\infty} \sqrt{p(x \mid \chi_i) p(x \mid \chi_j)} dx. \tag{4}$$

After a little algebra, the following for Gaussian distributions $\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\mu_j, \Sigma_j)$.

$${}_{CB} = \sqrt{P(\chi_i) P(\chi_j)} e^{(-B)}, \tag{5}$$

where

$$\begin{aligned} \epsilon_{CB} &= \sqrt{P(\omega_i) P(\omega_j)} \exp(-B) \\ B &= \frac{1}{8} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \left(\frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2} \ln \frac{\left| \frac{\Sigma_i + \Sigma_j}{2} \right|}{\sqrt{|\Sigma_i||\Sigma_j|}} \end{aligned} \tag{6}$$

and the determinant of the related matrix is denoted by $|.|$. The Bhattacharyya distance is a class separability measure that is defined by the letter B. Based on empirical research involving normal distributions, [37] proposes an equation that links the optimal Bayesian error with the Bhattacharyya distance. This was then used in [38] for feature selection.

We may iterate on the features and rank them using the aforesaid feature ranking test. For each channel, we will use the top four ranking features.

1. Ch1: RMS, crest factor, Std, kurtosis
2. Ch2: Mean, skewness, Std, RMS
3. Ch3: THD, crest factor, SNR, SINAD

The distributions for various features extracted from Ch1, Ch2, and Ch3 are visualised in Figure 3 and are divided by labels. The X-axis displays the value of the features for the Bhattacharya, Wilcoxon, and t-tests.

4. Background

4.1. Bidirectional LSTM

In 1997, Schuster and Paliwal [39] created the bidirectional recurrent neural network (BRNN), which connects the recurrent architecture, which has two hidden layers in the opposite direction to generate a result. This bidirectional tendency boosts the recurrent architecture's input data versatility. Furthermore, the recurrent bidirectional network improves inputs for the future state to the existing state's reachability and does not necessitate the fixation of input data before the training phase [40].

In this work, the recurrent unit of the bidirectional recurrent architecture was the long short-term memory (LSTM) because it avoids the problem of vanishing/exploding gradients that occurs in recurrent neural

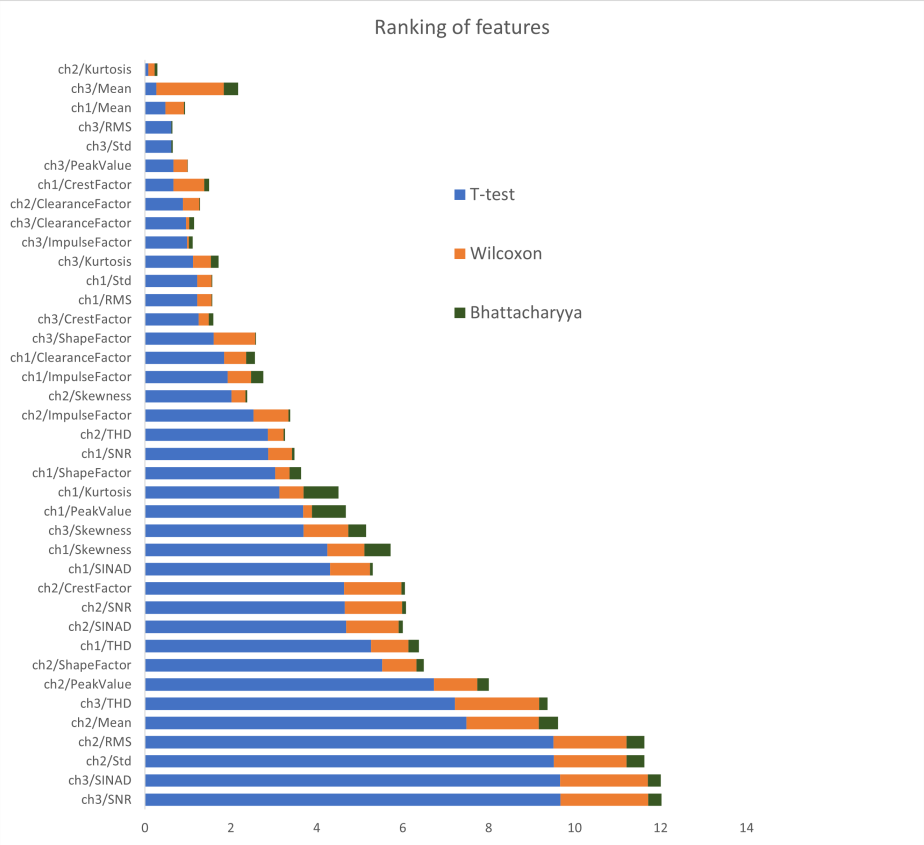


Figure 3. Features importance ranking.

networks (RNN). Graves et al. [41] also found that utilising the LSTM in a bidirectional architecture improved classification accuracy significantly. Figure 4a depicts the LSTM architecture, with h_t, C_t, and x_t representing the output of the LSTM at time t, the memory state cell, and input at the moment t. The symbol $\odot$ symbolises the multiplication of elements (Hadamard). The hyperbolic tangent function is $\tanh$, and the logistic sigmoid function is σ [42]. The values of the LSTM components are determined as follows:

$$i_t = \sigma\left(W_{xi}x_t + U_{hi}h_{t-1} + b_i\right) \tag{7}$$

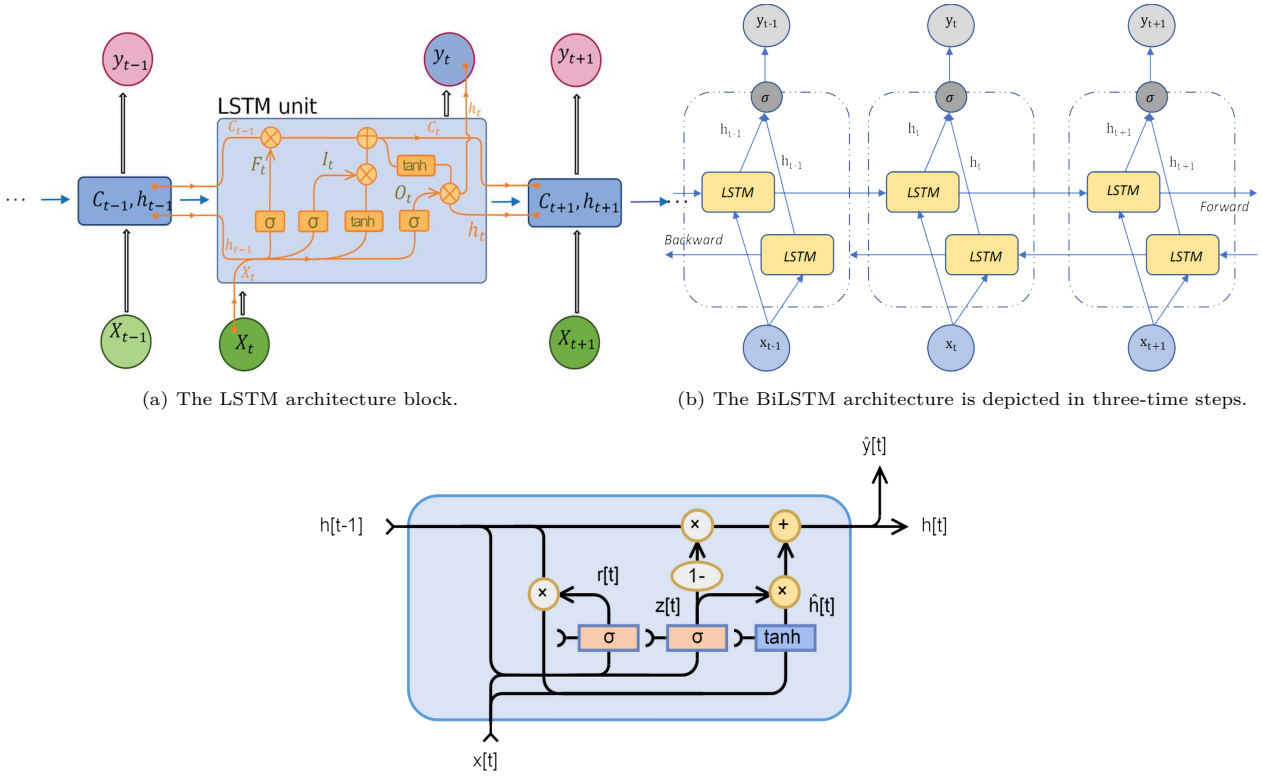
$$g_t = \tanh\left(W_{xg}x_t + U_{hg}h_{t-1} + b_g\right) \tag{8}$$

$$f_t = \sigma\left(W_{xf}x_t + U_{hf}h_{t-1} + b_f\right) \tag{9}$$

$$O_t = \sigma\left(W_{xo}x_t + U_{ho}h_{t-1} + b_o\right) \tag{10}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \tag{11}$$

$$h_t = \tanh\left(C_t\right) \odot O_t \tag{12}$$



(a) The LSTM architecture block.

(b) The BiLSTM architecture is depicted in three-time steps.

(c) Gated recurrent unit (GRU).

Figure 4. The LSTM architecture block attached to BiLSTM architecture and gated recurrent unit (GRU).

The input, forget, and output gates are denoted by i_t, f_t, and o_t respectively. The input-update value is g_t. The biases of each gate are b_i, b_a, b_f, and b_o. The feedforward weights are W, and the recurrent weights are U. Update of the input and activation of the output are the two activation units in the model. It is recommended that you employ the tanh activation function [43]. The activation function tanh is a saturating activation function. In RNNs, it is commonly employed as the recurrent activation function.

Figure 4b depicts the layout training process with BiLSTM, in which the Bi-LSTM algorithm calculates two hidden sequences: the $\overrightarrow{h}$ hidden layer (ahead) and $\overleftarrow{h}$ the hidden layer (backwards). Iterate the first layer increasing from time $t = 1$ to $t = T$ and the backward sequence layer decreasing from $t = T$ to $t = 1$ time to create the output sequence y [44]. The following formulas are used to calculate the output and forward/backward sequences:

$$y_t = W_{\vec{h}_y}\overrightarrow{h_t} + W_{\overleftarrow{h}_y}\overleftarrow{h_t} + b_y \tag{13}$$

$$\overrightarrow{h_t} = \mathcal{H}\left(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\overrightarrow{h_{t-1}} + b_{\vec{h}}\right) \tag{14}$$

$$\overleftarrow{h_t} = \mathcal{H}\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h_{t+1}} + b_{\overleftarrow{h}}\right) \tag{15}$$

The bidirectional structure takes into account the recurrent system's temporal dynamics as the model is fed both forward and backward [39].

4.2. Gated recurrent unit

In 2014, Cho et al. [45] introduced gated recurrent units (GRUs) as a method for gating recurrent neural networks. The GRU is akin to a forget gated long short-term memory (LSTM) as seen in Figure 4c, but it does not have a gate that controls output, so it has limited options. Modelling polyphonic music, modelling of speech signals, and activities using natural language processing, GRU's performance was shown to be on par with that of the LSTM in some circumstances. On datasets that are smaller and less common, GRUs have been proven to be more effective [46]. The values of the GRU neural network components are determined as follows:

Initially, the output vector for $t=0$ is $h_0=0$.

$$z_t = \sigma\left(W_{xi}x_t + U_{hi}h_{t-1} + b_i\right) \tag{16}$$

$$r_t = \sigma\left(W_{xf}x_t + U_{hf}h_{t-1} + b_f\right) \tag{17}$$

$$\hat{h}_t = \tanh\left(W_{xg}x_t + U_{hg}\left(r_t \odot h_{t-1}\right) + b_g\right) \tag{18}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{19}$$

Figure 5 depicts the suggested BiLSTM-GRU based anomaly detection model. There are nine layers in the suggested paradigm. The BiLSTM layer, which consists of 16 hidden units, minimizes the training parameters, total computational cost, and prevents the training model from overfitting. The GRU components have 32 hidden units come next, followed by BiLSTM layers. These layers are in charge of figuring out the network flow's temporal relationship and changing the temporal dynamics, owing to the GRU recurrent neural networks bidirectional design.

The loss function for this model autoencoder neural networks is MSE.

$$\text{Loss}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \widehat{x_i}\right)^2 \tag{20}$$

The loss function error is reduced by adjusting network structure parameters.

The benefit of this strategy over typical autoencoder neural models is that when the current hidden layer receives hidden info on the state from the prior time h_{t-1}, it has been added to the present time.

The hidden states are influenced not just by the most recent input x_t, but also by the info stored in a hidden state during the update. The network aids in the re-construction of present input data by retrieving their temporal features, and the use of timing dependence improves anomalous sample reconstruction accuracy, allowing for the generation of lengthy sequence abnormal samples.

5. Autoencoder model training

To identify if a machine is operating normally, we train a BiLSTM-GRU autoencoder. An autoencoder is trained on a set of 12 features taken from healthy operating data gathered promptly after a planned maintenance period using vibration sensors. MATLAB® was used to compute features from raw data. There are two labelled states in the data: before and after. Data acquired before and after maintenance is referred to as this. We will presume that the data gathered after maintenance reflects a typical (healthy) working condition. Because we were performing regular maintenance, we may not be able to claim the same for the previous data; this

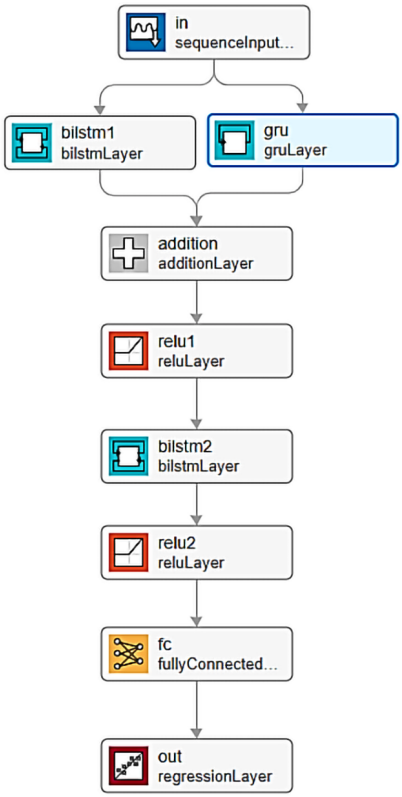


Figure 5. Proposed BiLSTM-GRU hyrbid model.

data could be normal or abnormal. For training, we choose 90% of healthy data out of 17,642 raw data. The proposed model was implemented on a Ryzen 5-3600 CPU and NVIDIA GTX 1650 super graphic card with Windows 10 OS. The full Training progress is depicted in Figure 6.

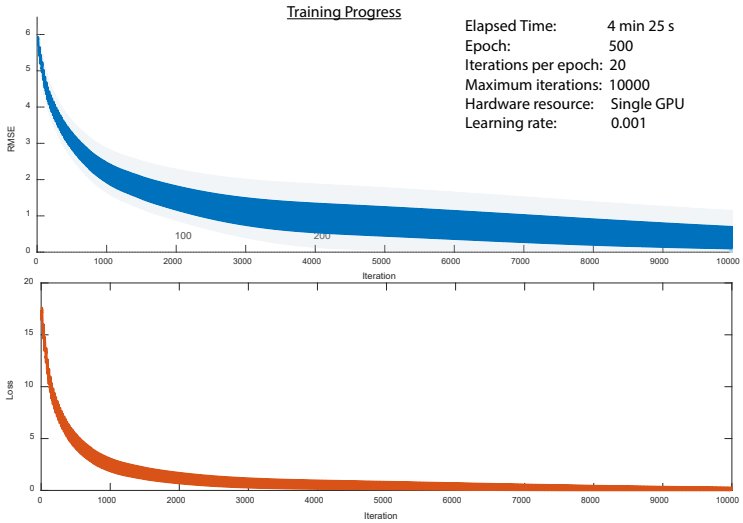
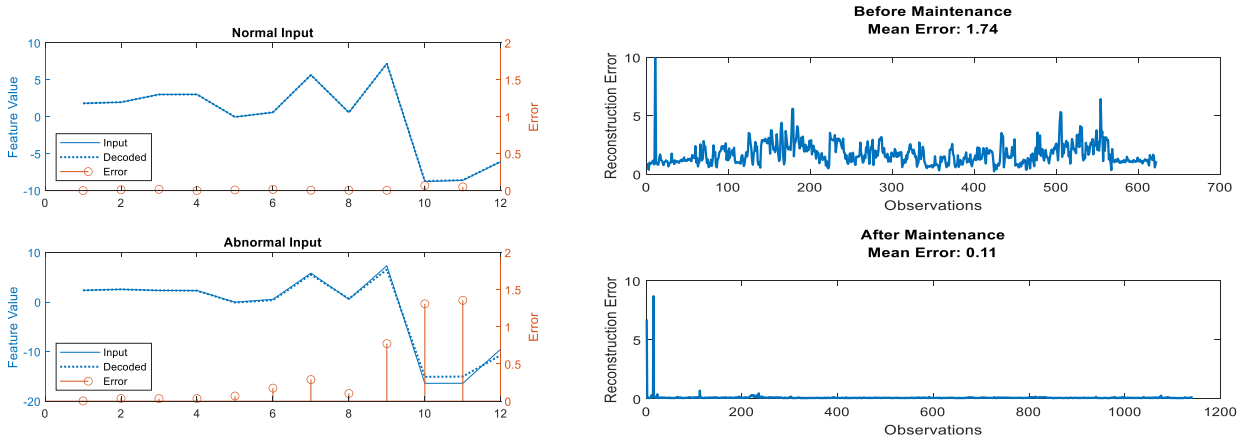


Figure 6. Training vs. validation loss for the proposed BiLSTM-GRU based model.

6. Experimental results and analysis

When the autoencoder receives data that does not appear to be healthy, it will have a tougher job reconstructing the signal. This will suggest an anomaly. A sample from the before and after maintenance should be extracted and visualized as in Figure 7a. Note that the charts below compare the error in each of the 12 characteristics' values (indicated on the x-axis). We can see that features 10–12 in this sample do not reconstruct adequately for the anomalous input, resulting in a significant error value. This could be a sign that something isn't quite right.



(a) Error in before and after maintenance of sampled data.

(b) Model error on validation data.

Figure 7. Error in before and after maintenance of sampled data, BiLSTM-GRU model error on validation data.

Figure 7b. shows the extract data of before and after maintenance using our suggested model. The reconstruction error for the data before maintenance is much higher than the data after maintenance, as seen in the graph. Because the autoencoder was trained on the data before maintenance, it will be able to reconstruct similar signals more accurately.

Four standard metrics are used to calculate the recall, precision, accuracy, and F1-score of the proposed BiLSTM-GRU model for anomaly detection of rotating equipment. Based on the operator's experience, we will set a 0.5 threshold limit for our machine. As a result, an anomaly is defined as a point with a reconstruction error that is 0.5 times larger than the mean across all observations.

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

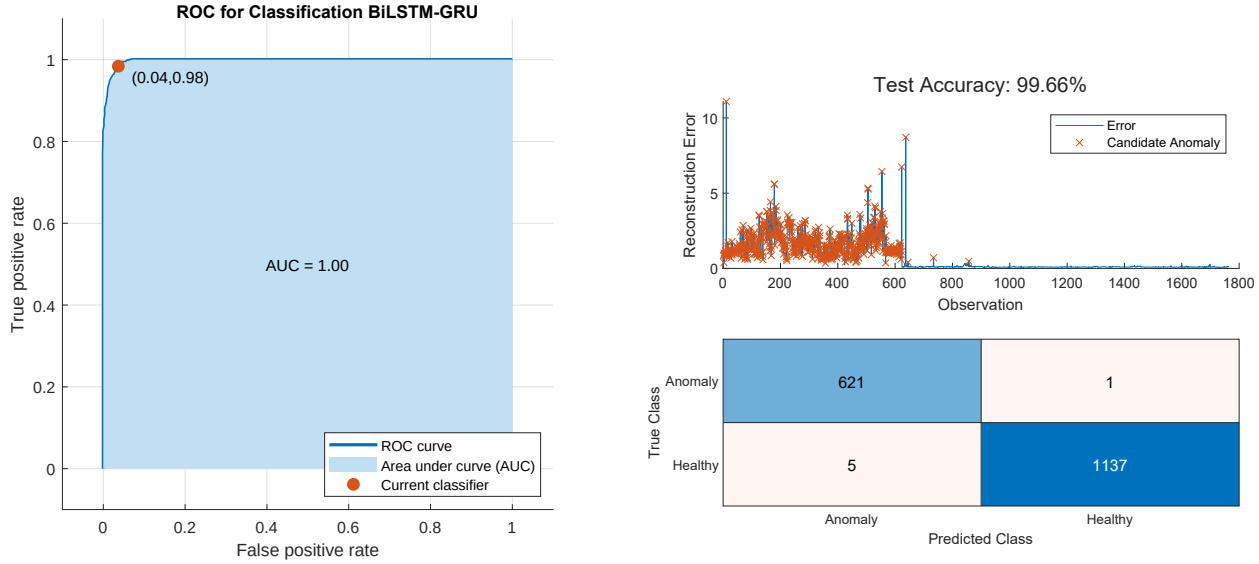
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

$$F_1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{24}$$

True positives, true negatives, false negatives, and false positives are represented by TP, TN, FN, and FP, respectively.

6.1. Discussion

We compared the performance of our proposed BiLSTM-GRU model to state-of-the-art machine learning models in detecting rotating machinery abnormalities. The ROC curve of the model is shown in Figure 8a. The receiver operator characteristic (ROC) curve is a tool for evaluating binary classification issues. It is a probability curve that compares the TPR (true positive rate) to the FPR (false positive rate) at various threshold values, allowing the 'signal' to be distinguished from the 'noise'. The area under the curve (AUC) is a summary of the ROC curve that measures a classifier's ability to distinguish between classes.



(a) ROC curve.

(b) Accuracy of the proposed model on 0.5 threshold limit.

Figure 8. ROC curve and accuracy of the proposed model on 0.5 threshold limit.

The following is a list of additional hyperparameters to consider. The number of epochs is 500, with a minibatch size of 500. The learning rate starts at 0.001, decays to 0.01, and the models are trained with a momentum of 0.9 using the adaptive moment estimation (Adam) optimizer. Figure 8b depicts the suggested model's confusion matrix and accuracy at the 0.5 threshold limit. The test is performed ten times to demonstrate that the proposed model is stable. Table 2a shows the average diagnosis performance of all studies. Many studies on anomaly diagnostic algorithms based on shallow and deep learning have been published in recent years. The suggested method is compared to recent approaches such as K-means [47], ANN [48], LSTM [49], RF-ET [50], and BiLSTM-RNN [51] to demonstrate its originality in the field of anomaly diagnosis. The average diagnostic accuracy of each technique is shown in Table 2b. When the results of the proposed strategy are compared to the results of other methods, it is obvious that the proposed strategy enhances diagnosis accuracy, proving its efficacy. Our suggested BiLSTM-GRU based model outperforms state-of-the-art anomaly detection methods, as shown in Table 2b.

7. Conclusion

For industrial rotating machinery, the suggested BiLSTM-GRU anomaly detection model outperforms state-of-the-art algorithms. It can also be implemented and used on any industrial machine. It can also be linked to an alert system that either controls machine anomalies or identifies any irregularities and sends a notification to

Table 2. The BILSTM-GRU empirical results and compared to other models.

The BILSTM-GRU empirical results.

Metrics	Testing outcome
Recall	99.84%
Precision	99.20%
Accuracy	99.66%
F1-score	99.52%
AUC	99.62%
Parameters of training	15,878
Total parameters	17642

Suggested BiLSTM-GRU compared to other models.

Model	Model methodology	Accuracy
Pahl et al.[47]	K-means	96.3%
Diro et al.[48]	ANN	98.27%
Azumah et al.[49]	LSTM	97.94%
Alrashdi et al.[50]	RF-ET	98.01%
McDermott et al.[51]	BiLSTM-RNN	98.48%
Our model	BiLSTM-GRU	99.66%

permitted members and takes necessary steps to mitigate the existing threat to maintain control of the situation and to protect their industrial machinery.

A method for ranking all the features extracted from the time-series data from a 3-axis accelerometer was also proposed. When considering variations in machine wear circumstances, BiLSTM-GRU anomaly detection model techniques could be used reliably in industrial use. In practice, the machine maker could calibrate our automatic technique once by upgrading their threshold limits, removing the requirement for continuous recalibration and hand-tuning by the machine operator. The methods given here were easy to apply to a real-time monitoring procedure, and the anomalies response was instantaneous.

During training, this model obtains optimal performance in a very short time. The broad component's memorising and regularisation capabilities can be increased in the future, allowing it to provide significantly better accuracy when paired with deep components. New reduction and feature crossing strategies can be developed to further improve the situation. Future research could include combining other regression models with additional deep neural network techniques.

References

[1] Chandola V, Banerjee A, Kumar V. Anomaly Detection: A Survey. Association for Computing Machinery 2009; 41 (3): PP. 1-58. doi: 10.1145/1541880.1541882

[2] Ariyaluran Habeeb RA, Nasaruddin F, Gani A, Targio Hashem IA, Ahmed E et al. Real-time big data processing for anomaly detection: A Survey. International Journal of Information Management 2019; 45: 289-307. doi: 10.1016/j.ijinfomgt.2018.08.006

[3] Chalapathy R, Chawla S. Deep Learning for Anomaly Detection: A Survey. 2019; doi: 10.48550/arXiv.1901.03407

[4] Fu T. A review on time series data mining. Engineering Applications of Artificial Intelligence 2011; 24 (1): 164-181. doi: 10.1016/j.engappai.2010.09.007

[5] Maharaj EA, D'Urso P. A coherence-based approach for the pattern recognition of time series. Physica A: Statistical Mechanics and its Applications 2010; 389 (17): 3516-3537. doi: 10.1016/J.PHYSA.2010.03.051

[6] Domańska D, Wojtylak M. Application of Fuzzy Time Series Models for Forecasting Pollution Concentrations. Expert Syst. Appl. 2012; 39 (9): 7673–7679. doi: 10.1016/j.eswa.2012.01.023

[7] Yang G, Yang H, Dai L. Time-series prediction modelling based on an efficient self-organization learning neural network. Application of fuzzy time series models for forecasting pollution concentrations 2015; 28 (8): 248-253. doi: 10.1016/J.IFACOL.2015.08.189

[8] Canizo M, Triguero I, Conde A, Onieva E. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. Neurocomputing 2019; 363: 246-260. doi: 10.1016/j.neucom.2019.07.034

[9] Hashemian HM, Bean WC. State-of-the-Art Predictive Maintenance Techniques. IEEE Transactions on Instrumentation and Measurement 2011; 60 (10): 3480-3492. doi: 10.1109/TIM.2009.2036347

[10] Theissler A. Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. Knowledge-Based Systems 2017; 123: 163-173. doi: 10.1016/j.knosys.2017.02.023

[11] Scime L, Beuth J. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. Additive Manufacturing 2018; 19: 114-126. doi: 10.1016/j.addma.2017.11.009

[12] Liu C, Ghosal S, Jiang Z, Sarkar S. An Unsupervised Spatiotemporal Graphical Modeling Approach to Anomaly Detection in Distributed CPS. In: ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS) 2016; PP. 1-10. doi: 10.1109/ICCPS.2016.7479069

[13] Angelo C, Luvisotto M, Michieletto G. Distributed Clustering Strategies in Industrial Wireless Sensor Networks. IEEE Transactions on Industrial Informatics 2017; 13 (1): 228-237. doi: 10.1109/TII.2016.2628409

[14] Goceri E. Skin Disease Diagnosis from Photographs Using Deep Learning. In: Lecture Notes in Computational Vision and Biomechanics 2019; 34: PP. 239-246. doi: 10.1007/978-3-030-32040-9-25

[15] Yaqoob I, Ahmed E, Hashem IAT, Abdalla ahmed AI, Gani AB et al. Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges. IEEE Wireless Communications 2017; 24 (3): 10-16. doi: 10.1109/MWC.2017.1600421

[16] Du S, Li T, Yang Y, Horng S. Multivariate time series forecasting via attention-based encoder–decoder framework. Neurocomputing 2020; 388: 269-279. doi: 10.1016/j.neucom.2019.12.118

[17] Triguero I, García-Gil D, Maillo J, Luengo J, García S et al. Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data. In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2019; 9 (2). doi: 10.1002/WIDM.1289

[18] Fernández A, García S, Galar M, Prati R, Krawczyk B et al. Learning from imbalanced data sets, Springer, Cham 2020, doi: 10.1007/978-3-319-98074-4

[19] Bayati A, Nguyen KK, Cheriet M. Multiple-Step-Ahead Traffic Prediction in High-Speed Networks. IEEE Communications Letters 2018; 22 (12): 2447-2450. doi: 10.1109/LCOMM.2018.2875747

[20] Li Z, Fang H, Huang M, Wei Y, Zhang L. Data-driven bearing fault identification using improved hidden Markov model and self-organizing map. Computers & Industrial Engineering 2018; 116: 37-46. doi: 10.1016/j.cie.2017.12.002

[21] Bigdeli E, Raahemi B, Mohammadi M, Matwin S. A fast noise resilient anomaly detection using GMM-based collective labelling. In: 2015 Science and Information Conference (SAI); 2015. pp. 337-344. doi: 10.1109/SAI.2015.7237166

[22] Xiang P, Zhou H, Li H, Song S, Tan W et al. Hyperspectral anomaly detection by local joint subspace process and support vector machine. International Journal of Remote Sensing 2020; 41 (10): 3798-3819. doi: 10.1080/01431161.2019.1708504

[23] Zhou P, Li Z, Snowling S, Baetz BW, Na D et al. A random forest model for inflow prediction at wastewater treatment plants. Stochastic Environmental Research and Risk Assessment 2019; 33 (10): 1781-1792. doi: 10.1007/S00477-019-01732-9

[24] Ieracitano C, Adeel A, Morabito FC, Hussain A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing 2020; 387: 51-62. doi: 10.1016/j.neucom.2019.11.016

[25] Hu X, Dai J, Huang Y, Yang H, Zhang L et al. A weakly supervised framework for abnormal behavior detection and localization in crowded scenes. Neurocomputing 2020; 383: 270-281. doi: 10.1016/j.neucom.2019.11.087

[26] Singh K, Rajora S, Vishwakarma DK, Tripathi G, Kumar S et al. Crowd anomaly detection using Aggregation of Ensembles of fine-tuned ConvNets. Neurocomputing 2020; 371: 188-198. doi: 10.1016/j.neucom.2019.08.059

[27] Fan J, Zhang Q, Zhu J, Zhang M, Yang Z et al. Robust deep auto-encoding Gaussian process regression for unsupervised anomaly detection. Neurocomputing 2020; 376: 180-190. doi: 10.1016/j.neucom.2019.09.078

[28] Che Z, Purushotham S, Cho K, Sontag D, Liu Y. Recurrent Neural Networks for Multivariate Time Series with Missing Values. Scientific Reports 2018; 8 (1): 1-12. doi: 10.1038/s41598-018-24271-9

[29] Malhotra P, Vig L, Shroff G, Agarwal P. Long short term memory networks for anomaly detection in time series. In: 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2015, Bruges, Belgium

[30] Nguyen HD, Tran KP, Thomassey S, Hamad M. Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. International Journal of Information Management 2021; 57: 102282. doi: 10.1016/j.ijinfomgt.2020.102282

[31] Wang M, Wang Z, Lu J, Lin J, Wang Z. E-LSTM: An Efficient Hardware Architecture for Long Short-Term Memory. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 2019; 9 (2): 280-291. doi: 10.1109/JETCAS.2019.2911739

[32] Jiang K, Wang W, Wang A, Wu H. Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network. IEEE Access 2020; 8: 32464-32476. doi: 10.1109/ACCESS.2020.2973730

[33] Yuan J, Tian Y. An Intelligent Fault Diagnosis Method Using GRU Neural Network towards Sequential Data in Dynamic Processes. Processes 2019; 7: 152-155. doi: 10.3390/PR7030152

[34] Afrasiabi S, Afrasiabi M, Parang B, Mohammadi M. Designing a composite deep learning based differential protection scheme of power transformers. Applied Soft Computing Journal 2020; 87: 105975. doi: 10.1016/J.ASOC.2019.105975

[35] Kimotho JK, Sextro W. An approach for feature extraction and selection from non-trending data for machinery prognosis. In: PHM Society European Conference 2014; 2: 1. doi: 10.36001/PHME.2014.V2I1.1462

[36] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations 2015. doi: 10.48550/arXiv.1412.6980

[37] Lee C, Choi E. Bayes error evaluation of the Gaussian ML classifier. IEEE Transactions on Geoscience and Remote Sensing 2000; 38 (3): 1471-1475. doi: 10.1109/36.843045

[38] Choi E, Lee C. Feature extraction based on the Bhattacharyya distance. Pattern Recognition 2003; 36 (8): 1703-1709. doi: 10.1016/S0031-3203(03)00035-9

[39] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 1997; 45 (11): 2673-2681. doi: 10.1109/78.650093

[40] Salehinejad S, Sankar S, Barfett J, Colak E, Valaee S. Recent Advances in Recurrent Neural Networks. Neural and Evolutionary Computing 2021. doi: 10.48550/arXiv.1801.01078

[41] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 2005; 18 (5): 602-610. doi: 10.1016/j.neunet.2005.06.042

[42] Jain LC, Medsker LR. Recurrent Neural Networks: Design and Applications. USA: CRC 1999

[43] Elsayed N, Maida A, Bayoumi M. Effects of different activation functions for unsupervised convolutional LSTM spatiotemporal learning. Advances in Science, Technology and Engineering Systems 2019; 4 (2): 260-269. doi: 10.25046/AJ040234

[44] Graves A, Jaitly N, Mohamed AR. Hybrid speech recognition with Deep Bidirectional LSTM. In: IEEE Workshop on Automatic Speech Recognition and Understanding 2013; 273-278. doi: 10.1109/ASRU.2013.6707742

[45] Cho K, Merrienboer BV, Gulcehre C, Bahdanau D, Bougares F et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Computation and Language 2014. doi: 10.48550/arXiv.1406.1078

[46] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Neural and Evolutionary Computing 2014; 18 (5): 602-610. doi: 10.48550/arXiv.1412.3555

[47] Pahl M, Aubet F. All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection. 14th International Conference on Network and Service Management (CNSM), 2018, pp. 72-80.

[48] Diro AA, Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems 2018; 82: 761-768. doi: 10.1016/J.FUTURE.2017.08.043

[49] Azumah SW, Elsayed N, Adewopo V, Zaghloul S, Li C. A deep lstm based approach for intrusion detection iot devices network in smart home. In: IEEE 7th World Forum on Internet of Things (WF-IoT) 2021; 836-841. doi: 10.1109/WF-IoT51360.2021.9596033

[50] Alrashdi I, Alqazzaz A, Aloufi E, Alharthi R, Zohdy M et al. AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. In: IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) 2019;pp. 0305-0310. doi: 10.1109/CCWC.2019.8666450.

[51] McDermott CD, Majdani F, Petrovski AV. Botnet Detection in the Internet of Things using Deep Learning Approaches. In: International Joint Conference on Neural Networks (IJCNN) 2018;pp. 1-8. doi: 10.1109/IJCNN.2018.8489489.