Bearing Anomaly Detection in an Air Compressor using an LSTM and RNN-Based Machine Learning Model

Byung Gun Joung

bjoung@purdue.edu

Purdue University https://orcid.org/0000-0003-0992-7534

Chandra Nath

Zhongtian Li

John W. Sutherland

Research Article

Keywords: predictive maintenance, machine learning, machine health management, vibration monitoring

Posted Date: April 30th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4298970/v1

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on September 4th, 2024. See the published version at https://doi.org/10.1007/s00170-024-14322-z.

Bearing Anomaly Detection in an Air Compressor using an LSTM and RNN-Based Machine Learning Model

Byung Gun Joung*, Chandra Nath, Zhongtian Li, John W. Sutherland

School of Environmental and Ecological Engineering
Purdue University, West Lafayette, IN 47907-2088, USA

*Corresponding author. *E-mail address:* bjoung@purdue.edu

Abstract
Smart systems such as data-driven machine health monitoring are emerging as a powerful technology for advanced manufacturing as a result of the availability of low-cost sensors, wireless communication, and advances in computational capabilities. Machine learning represents a critical element of manufacturing equipment health monitoring due to its ability to efficiently process large amounts of data and move beyond traditional rule-based maintenance methods. Predictive maintenance (PdM) has become increasingly popular in manufacturing. PdM can detect faults, determine root causes of operation anomalies, estimate the current health state of a system, and predict the future state and time when a component will fail in the absence of an intervention. One weakness of many past studies is the lack of run-to-failure data from an actual production environment. This paper presents run-to-failure data for the air compressor of an injection molding machine. A Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) is proposed to detect bearing faults in the air compressor. The model achieves a 97.4% of prediction accuracy (95.3% of overall accuracy). Experiments for machine state classification are also conducted and the classification performance compares favorably with conventional models.

Keywords: predictive maintenance, machine learning, machine health management, vibration monitoring

NOMENCLATURE

DTFT	Discrete Time Fourier Transform
FFT	Fast Fourier Transform
STFT	Short Time Fourier Transform
HVAC	heating, ventilation, and air conditioning
IIoT	Industrial Internet of Things
CNN	Convolutional Neural Network
KNN	K-Nearest Neighbor
OCSVM	One-Class Support Vector Machine
AE	Autoencoder
RF	Random Forest

1. Introduction

Predictive maintenance (PdM) has become increasingly popular in the industrial sector as a means of managing maintenance activities based on the condition of machine tools, equipment, and systems [1]. PdM can prevent unexpected failures and provide significant advantages in relation to product quality, safety, machine availability, and cost reduction by predicting faults, determining root causes of operation anomalies, estimating the current health state of a system, and forecasting the future state and remaining useful life of a system [2], [3]. It is widely regarded as a data-driven analytical application for Industry 4.0, requiring appropriate implementation of sensors, data collection, storage, analysis, and visualization. The implementation of PdM in manufacturing requires careful consideration of machine specifications and manufacturing requirements [4]. A key factor of implementing PdM is the prioritization of manufacturing assets or machine tools that are critical to the manufacturing activity. To quantify the criticality of the manufacturing equipment, machine-specific knowledge and process domain knowledge are required to determine machine-specific failure signals [5]. Effective feature extraction procedures and data analysis techniques are also important to extract signals indicating incipient failure, detect and diagnose fault type, and suggest maintenance interventions for different manufacturing scenarios [6]. The Industrial Internet of Things (IIoT) and data analytics have enabled the collection and processing of high-volume machine state data for decision-making in PdM [7]. The development of predictive algorithms using deep neural networks has created opportunities for identifying emerging anomalies. These algorithms have shown their ability to handle high-dimensional data with feature extraction, reduce data dimensionality for knowledge abstraction, and identify process relations; the applicability of these techniques in various industrial applications have also been demonstrated [8].

One of the key benefits of predictive maintenance is that it can reduce downtime, saving businesses significant amounts of money in the long term. By monitoring the health of machinery and systems, maintenance can be carried out when required, rather than relying on regular, scheduled maintenance that may not be necessary. This can also help to extend the life of machines, by detecting potential problems early and fixing them before they become major issues. PdM also plays an important role in ensuring safety in industrial settings. By identifying faults and predicting approaching failures, PdM can help to prevent productivity loss/accidents caused by machine failure. This is especially important in industries such as manufacturing, where machinery is often operating at high speeds and under extreme conditions. In addition to safety and cost savings, predictive maintenance can also improve product quality. By identifying and fixing potential problems before they impact the production process, manufacturers can ensure that their products meet the required specifications and standards.

A common piece of equipment found in many manufacturing facilities is an air compressor. Compressed air is important for many manufacturing processes, e.g., pneumatic systems for device actuation and process cooling [9]. Air compressors play an important part in the energy efficiency and productivity of a manufacturing facility. Reciprocating and rotary-screw, are two main types of compressors to be used in industries. A reciprocating compressor uses pistons to produce compressed air [10], and crew compressors are widely used as high-efficiency and compact gas compression equipment. However, due to the positive displacement nature of a screw compressor, the screw may experience excessive wear, especially in a high particulate matter environment.

Early detection of abnormal behavior by condition monitoring (e.g., vibration monitoring) could provide indication of damage in mechanical components of the machine, so that operators can timely schedule maintenance [11].

In this paper, an anomaly detection algorithm is proposed to categorize the condition of a machine into different states (stop, normal, near failure, failure). A case study used in this paper considers two air compressors at a pharmaceutical manufacturing company; one of the compressors is primarily used and the other is used as a back-up. The primary air compressor has been monitored for nearly 2 years and bearing failure has been identified by the proposed anomaly detection model as well as the maintenance engineer.

Anomaly detection is a type of unsupervised learning technique that can identify outliers or unknown patterns by modeling the density of data. There are three main categories of anomaly detection methods: statistical, distance-based, and clustering-based. For example, Principal Component Analysis (PCA) is a statistical-based method that uses Hotelling's T^2 and SPE statistics based on a predefined confidence level [12]. The K-Nearest Neighbor (KNN) algorithm is a distance-based method that computes the anomaly score of each data sample based on the average distance to its k nearest neighbors. Clustering-based methods model different data clusters and find anomalies via a predefined outlier score, such as the One-Class Support Vector Machine (OCSVM) [13], which is widely used for outlier detection and estimates the density distribution of data to identify unseen data as normal or abnormal. However, these methods may not work well on multivariate time series data since they cannot capture the temporal dependencies appropriately. To address this issue, many studies have been conducted on deep learning-based models. In early-stage studies, two-step approaches were widely adopted to address the high dimensionality of input datasets, where the first step is dimensionality reduction followed by density estimation (e.g., Kernel Density Estimation) as the second step. However, this approach may lead to the loss of key information for anomaly detection and suboptimal anomaly scores. To overcome these limitations, pre-trained models such as VGG and ResNet have been adopted for learning feature representation based on the assumption that features extracted from such pre-trained models can preserve the discriminative information for anomaly detection. However, this approach has most often been applied to certain types of data, e.g., images and designed for large-scale applications.

An autoencoder (AE) is a powerful model for anomaly detection as it is capable of learning a normality feature in low-dimensional space and then reconstructing the data into the original space Gaussian mixture. The Deep Autoencoding Gaussian Mixture Model (DAGMM) aims to model the density distribution of multi-dimensional data using the Gaussian mixture model while minimizing the loss of the key information in a low-dimensional space [14]. A distance-based detection method such as Euclidean distance is used in the compression network of DAGMM. Other variants of AE such as Dense-AE and Long-Short Term Memory-based AE (LSTM-AE) utilize the Reconstruction Error (RE) for anomaly detection.

To leverage anomaly detection techniques for the manufacturing domain, a careful adaptation of prior knowledge about the input data is required to construct an optimal detection method for anomalies. Several methods utilized for machine diagnosis were reported, where different approaches are applied to construct anomaly information. Benedetti et al. proposed a statistic-based anomaly detection method using a simple neural network architecture to provide predictive alerts based on bounded daily residuals [15]. Their approach focuses on scoring anomalies and utilizing anomaly information to improve data-driven maintenance strategy. Inspired by the anomaly scoring method, an LSTM-RNN based anomaly detection algorithm is proposed and a case study is considered to investigate the detection capability of the proposed model in a real-world environment.

In the following section, the experimental environment and experimental methods will be described. For example, the sensor installation, data acquisition process, and data sampling methods for the work reported in the paper will be presented. The proposed LSTM-RNN based anomaly detection model will be discussed in section 3. The results of the proposed model are shown and discussed in section 4. Additional analysis to explore the effects of model hyperparameters such as learning rate, batch size, training/testing ratio of dataset is conducted and described in section 5.

2. Experiment Setup

An air compressor unit (CSD – T, KAESAR) which provides compressed air for an injection molding machine (JOMAR) for pharmaceutical manufacturing is monitored for the experiment. The machine is controlled by a variable speed driver (VSD) and operates at different speeds according to its operating mode (idle, half load, full load, off). Vibration behavior of the air compressor is measured using an accelerometer. The following section describes the data acquisition system and methodology for data collection.

2.1 Data Acquisition System

Mostly, smart manufacturing technologies are implemented into existing production systems[16], where different production subsystems may have different data communication interfaces. However, these systems often feature diverse production subsystems equipped with different data communication interfaces. The presence of non-standardized interfaces can potentially lead to security vulnerabilities and hinder interoperability with other networks or subsystems [17]. To address these challenges, many organizations are turning to open-source and edge device-based networks for conducting various data communication and processing tasks across different interfaces.

One such edge computing device gaining popularity is the Raspberry Pi, known for its cost-effectiveness, flexibility, and interoperability [18]. In the context of the present study, Raspberry Pi serves as the execution platform for the proposed anomaly detection model. It analyzes incoming data streams to identify anomalies and transmits relevant outcomes, such as threshold values, anomaly scores, and detected anomalies, to a centralized database for visualization purposes.

Python scripting emerges as a preferred choice for orchestrating actions related to data collection, pre-processing, and analysis in such scenarios . Leveraging the versatility of Python, researchers can develop customized scripts tailored to specific data processing requirements. In the current case, a Python script facilitates seamless integration of data acquisition and analysis tasks on the Raspberry Pi platform.

For data storage and management, a MySQL database is employed due to its compatibility with a wide range of web-based applications, including visualization software [19]. Additionally, cloud services play a crucial role in storing measurement data, such as raw acceleration data, for future analysis. This setup allows for efficient data retrieval and sharing across different devices and locations. Moreover, local data analysis devices, such as personal computers, can access the cloud repository to process raw data and facilitate model development.

In terms of sensor technology, vibration and sound signals serve as common indicators of mechanical failures in reciprocating machines. In this study, an accelerometer is utilized to monitor the vibration patterns of the air compressor. Figure 1 illustrates the comprehensive data acquisition system developed to support this research endeavor, showcasing the interconnection of various components.

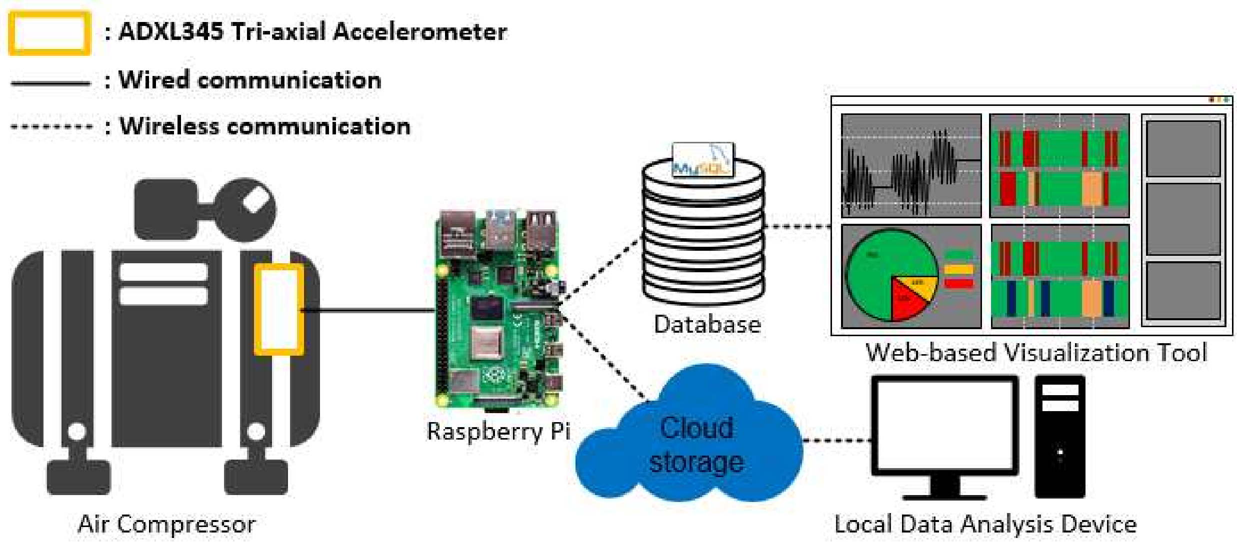


Fig. 1 Implementation of data acquisition system for air compressor

2.2 Measurement and Data Collection

The utilization of sensor technology plays a pivotal role in the monitoring and analysis of machinery performance, as depicted in Fig. 1. In this study, an ADXL345 tri-axial accelerometer was chosen to measure the vibration patterns of the air compressor. The selection of this sensor was based not only on its cost-effectiveness but also on its ability to sample data at a frequency that adequately covers the typical operating frequency range of the target machine, namely the air compressor. It is well established that the sampling frequency of a sensor should exceed at least twice the highest frequency component of the target signal to prevent signal distortion or loss. Hence, the ADXL345's sampling frequency was deemed sufficient for capturing the intricate vibration patterns of the air compressor.

To ensure comprehensive data collection aligned with the machine's supervisory control system process, vibration data was sampled at 15-minute intervals. During each sampling interval, an extensive dataset comprising 10,000 data points for each axis (X, Y, and Z) was collected at a sampling rate of 3,200 Hz. This meticulous data collection strategy enabled the capture of vibration patterns and fluctuations in the machine's operational behavior over time.

2.3 Data Preprocess

The flow diagram depicted in Fig. 2 shows the process for offline monitoring and data preprocessing. Following data collection, the time-domain signals were transformed into the time-frequency domain using a Short Time Fourier Transform (STFT). This transformation facilitated a deeper understanding of the frequency characteristics embedded within the vibration signals [20]. Specifically, the roughly 3.125 seconds worth of data for each axis (derived from 10,000 data points sampled at 3,200 Hz) were broken into 36 segments, and an STFT analysis was performed on each segment. The resulting frequency spectra provided valuable insights into the dynamic changes in frequency content occurring over the duration of data collection.

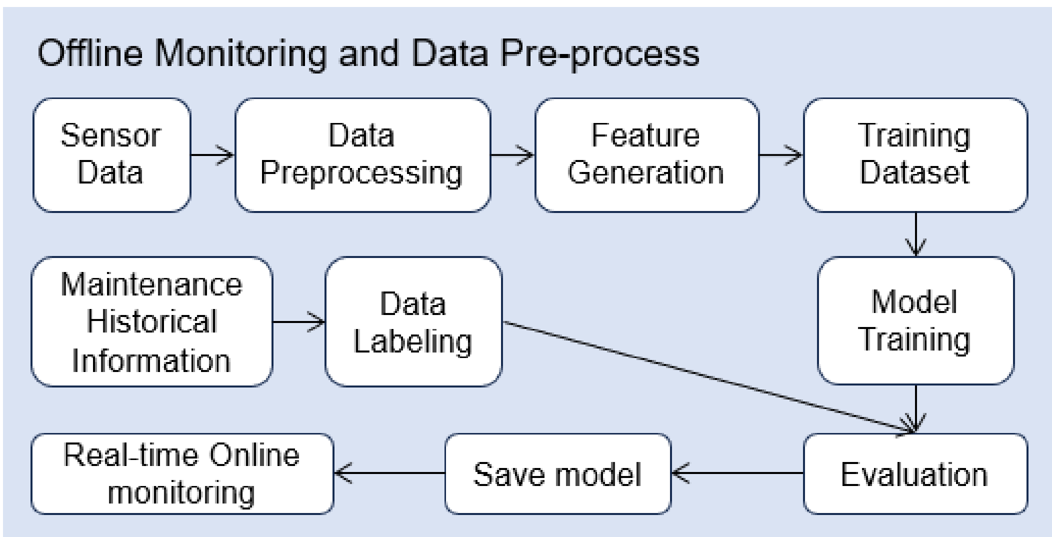
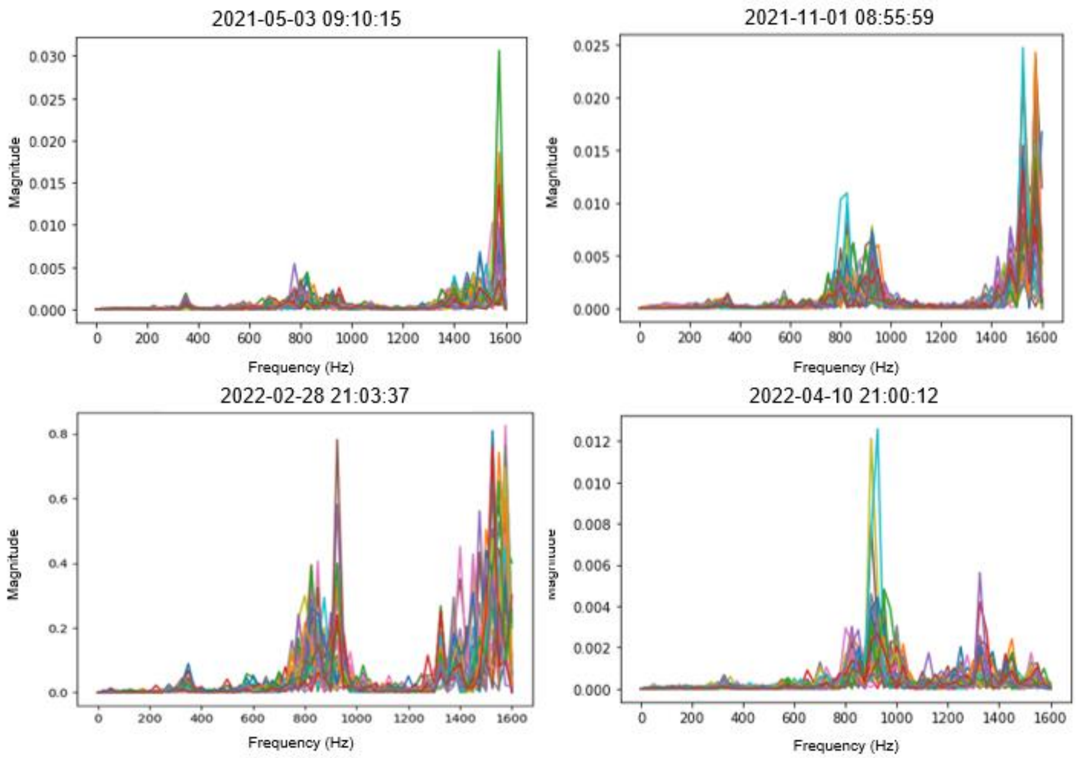
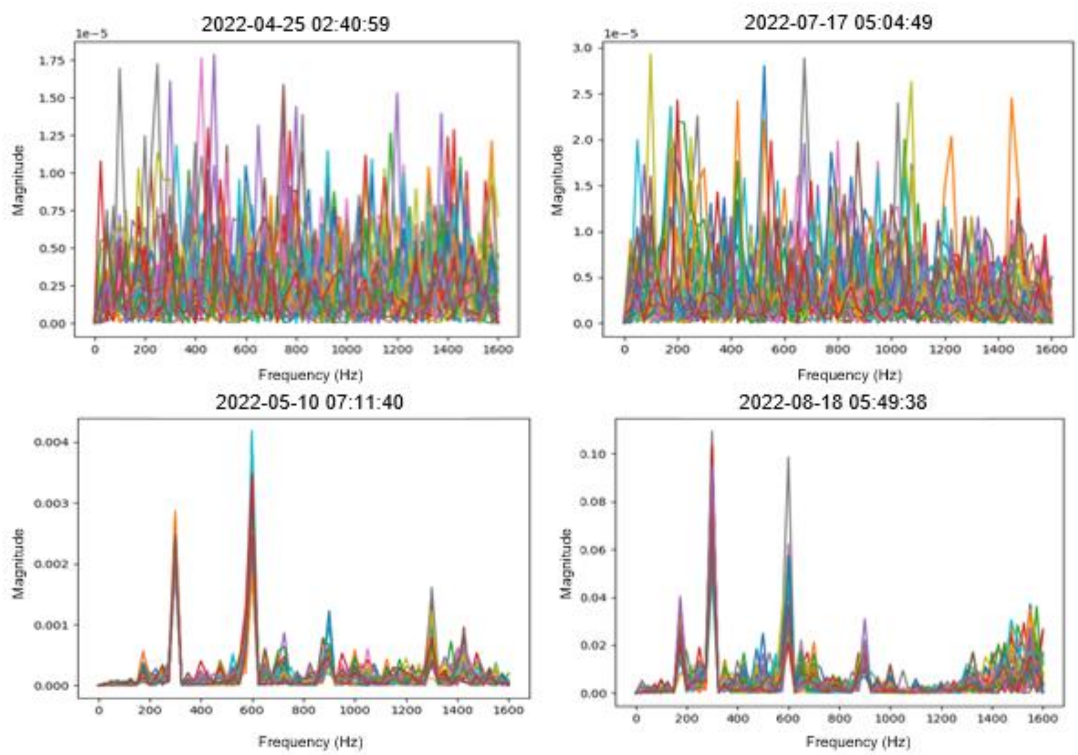


Fig. 2 Flow diagram of offline monitoring and data preprocessing

Concurrently, historical information on maintenance activities is used for data labeling, which is crucial for classification. These steps are followed by feature generation, where relevant features are extracted from the preprocessed data. The resulting features are compiled into a training dataset used for model training. After the model is trained, it is evaluated by comparing the model outcomes and the true labels. If the model's performance is satisfactory, it is saved for real-time online monitoring, where it can be used to make predictions or monitor system performance in an operational setting.



(a)



(b)

Fig. 3 Frequency spectra for x-axis (a) normal and near failure state (b) stop and failure state

Fig. 3 showcases examples of frequency domain vibration data for the x-axis acceleration signal which is collected over a span of nearly 2 years. Each plot within the figure illustrates the frequency spectra corresponding to the 36 different time segments (a different color for each segment), effectively highlighting the evolving frequency characteristics observed during the 3-second data collection window. This comprehensive analysis underscores the effectiveness of employing frequency domain techniques in discerning subtle changes and patterns within vibration signals, thereby facilitating informed decision-making and predictive maintenance strategies in industrial settings.

Let us now consider the behavior of the frequency spectra over time in Fig. 3. The plot on the top of Fig. 3 (a) is for a compressor operating normally (the time when the data was collected in shown above the graph). As is evident, the magnitude associated with different frequencies changes slowly as a function of time (time evolves from top-left to top-right to bottom-left to bottom-right). The two plots in the top portion of the figure are identified as being in a normal state and the two plots in the bottom portion of the figure may be identified as abnormal. Also, examples of frequency spectra of stop and failure state are depicted in Fig. 3 (b). For each data sample, the frequency spectra in the X, Y, and Z directions were assigned a label of 0, 1, 2, and 3, where 0 indicates the machine is stopped, 1 indicates a normal running state of the machine, 2 indicates the machine is near failure, and 3 indicates the machine is in failure. The labeling process was performed by ensuring that all labels are aligned with the supervisory control system process data. In total, 46,115 samples were collected and used for training and testing (with each sample consisting of 10,000 points collected for each axis, i.e., the X, Y, and Z directions). All samples were manually labeled based on the interpretation of vibration patterns.

3. Model Description

With data available from the vibration of the air compressor during injection molding machine operation, attention now shifts to the machine learning model that was utilized. Different from fundamental feedforward Deep Neural Networks (DNNs), an LSTM RNN model is able to model the temporal dynamics using some form of memory. Given an input sequence $X = (x_1, x_2, ..., x_T)$, an LSTM RNN layer computes the hidden states $H = (h_1, h_2, ..., h_T)$, iteratively via W_{oc}. Here, the matrix M contains all the parameters of the RNN and H is a nonlinear function. In the LSTM RNN architecture, the nonlinear H function is calculated using the following equations (Eqs. (1) – (5)).

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \tag{2}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \tag{4}$$

$$h_t = \phi(c_t) \odot o_t \tag{5}$$

Here, $\odot$ denotes element-wise multiplication, ϕ is a nonlinear function which transforms the multi-dimensional input to a single dimension on the range of: [−1, 1]; c_t denotes the states of the memory cells; i_t, f_t, o_t are the input gates, forget gates, and output gates of the memory cells, respectively. W_{ic}, W_{fc} and W_{oc} are the peephole connection matrices, which are usually diagonal.

The model that has been implemented takes inspiration from two existing networks, the dilated LSTM RNN network and the Temporal Hierarchical One Class (THOC) network [21], both of which address the issue of complex dependencies and vanishing gradients in layers. As vibration data collected from a machine is a time-series, an LSTM-RNN based layer has great potential to identify changes in vibration patterns over time. The THOC network has been modified to develop the proposed anomaly detection model, which is structured as shown in Fig. 3. It comprises three long-short term memory (LSTM) layers with dilated skip connections that have been implemented to capture different temporal dependencies and prevent vanishing and exploding gradient problems. The dilation values assigned to the layers were specifically chosen as 1, 3, and 6, respectively. This decision aimed to capture temporal dependencies occurring at intervals of 1, 3, and 6 time steps. The input sequence is initially processed by the 1st layer, with subsequent layers receiving inputs from specific elements of the output sequence. For instance, the elements at positions 1, 4, 7, and so forth, of the output from the 1st layer are fed into the 2nd layer. Similarly, alternating elements of the output from the 2nd layer are passed to the 3rd layer. This pattern continues, with each subsequent layer receiving inputs based on the temporal spacing corresponding to the initial input sequence (e.g., 1st, 7th, 13th time steps). These particular time step intervals were determined based on the model initialization, which exhibited optimal performance with dilation values of 1, 3, and 6. It's worth noting that different dilation values can have a significant impact on the model's performance. Larger dilation values have the capacity to capture longer temporal dependencies present in the sampled data.

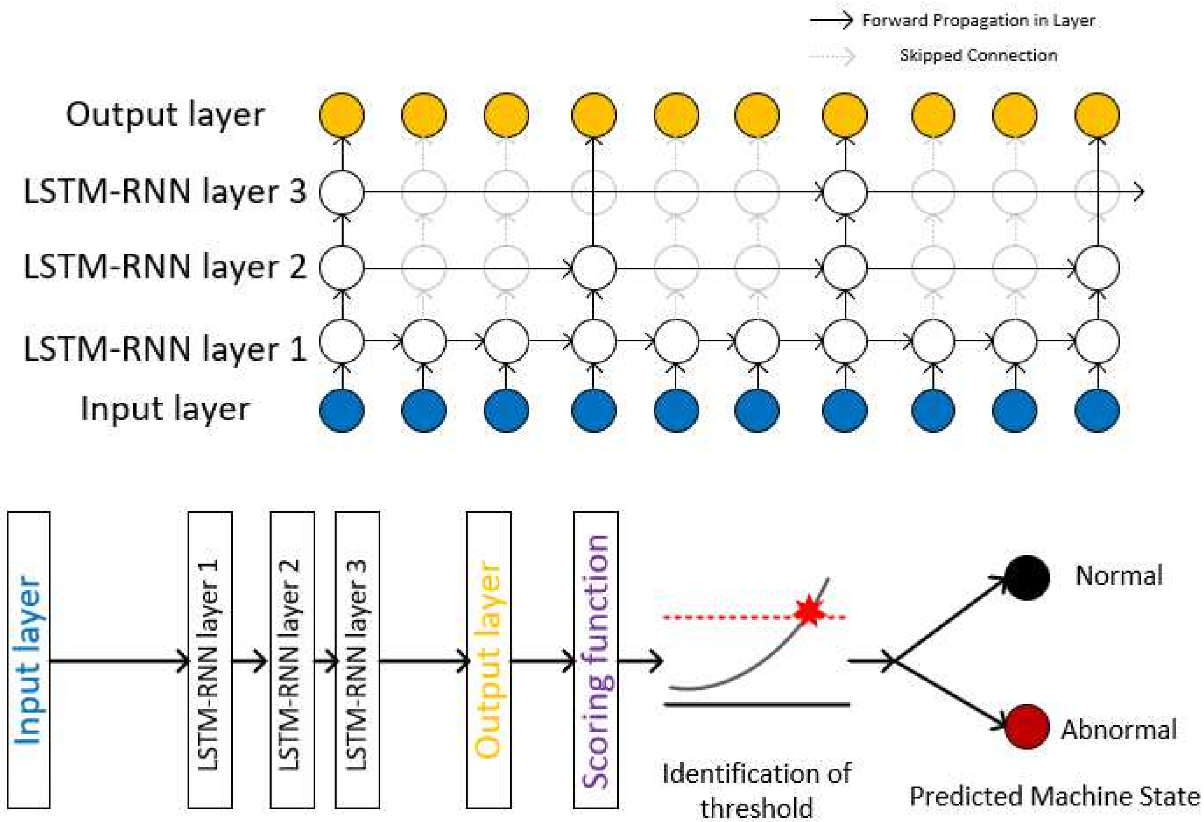


Fig. 4 Architecture of the proposed anomaly detection and anomaly detection process

In Fig. 4, each node represents a hidden RNN cell which can be described as:

$$c_t^{\,l} = f^{\,l}(x_t^{\,l}, c_{t-1}^{\,l}, c_{t-2}^{\,l}, \ldots) \tag{6}$$

where c_t^l refers to the RNN cell for layer l at time t. The number of input nodes (highlighted in blue in the figure) is determined by the length of the input data, which for the present case is 36. This number can also be interpreted as the number of time steps in the input data. The hidden state h_t of an RNN cell at time step t is computed based on the current input x_t and the previous hidden state h_{t-1} as follows.

$$h_t = tanh(W_{ih}x_t + b_{ih} + W_{hh}h_t - 1 + b_{hh}) \tag{7}$$

Where W_{ih} and W_{hh} are weight matrices for the input to hidden and hidden to hidden connections, respectively, and b_{ih} and b_{hh} are the corresponding bias terms. Each input cell has 195 frequency features as an input at time step t (3 axes with 65 spectrum values each). Such features are then utilized to calculate the similarity score between each feature and the number of clusters in each layer. Here, the number of clusters was predefined as 8, 4, and 4 respectively. These clusters are initialized by the first few training data with a KNN algorithm before training. During the training, the clusters in each layer will be updated. Euclidean distance is used as the score function. In general, different score functions may need to be evaluated since score function has a significant impact on the model performance. The output of the RNN cell can then be passed through a dense layer to produce the final output y_t by the following equation.

$$y_t = soft\max(W_{hy}h_t + b_{hy}) \tag{8}$$

The output (similarity score) obtained from the score function is translated to loss using equation 9; the loss will be minimized through the training process,

$$loss = \lambda(1 - d_{euclidean}(f_t^l, c_t^l)) \tag{9}$$

where λ is a weight parameter to penalize the impact of outliers and $d_{euclidean}$ is the Euclidean distance between input features and the clusters. The Adaptive Moment Estimation (Adam) optimization method was employed as it is known to have good capabilities of reaching global minimum with high dimensional datasets. The obtained anomaly scores are then normalized using a min-max scaler to classify anomalies based on a threshold value that will be predefined. As the LSTM layers with different dilation parameters can capture the different temporal dependencies, the model is expected to identify underlying temporal patterns from the data. A diverse combination of dilations can be used to explore the impact of the dilation value on the model performance.

The LSTM layer takes sequential input data obtained from the transformation of raw data from the time to frequency domain using the Short Time Fourier Transform (STFT). The transformed data contains 65 frequency components for each axis representing mechanical vibration frequency components. These frequency features are obtained by splitting the input time series into 36 segments with 64 overlapping data points. 65 frequency features are obtained for each axis after the STFT. Such frequency features for 3 axis are then stacked in order to obtain 195 frequency features. 195 frequency features are used as input data. In the LSTM layers, each input layer node takes 195 frequency features of each time step. The output gate of each node passes the updated parameters to the adjacent node, which are the 2nd node in the first LSTM layer, 4th node in the second layer, and 7th node in the third layer.

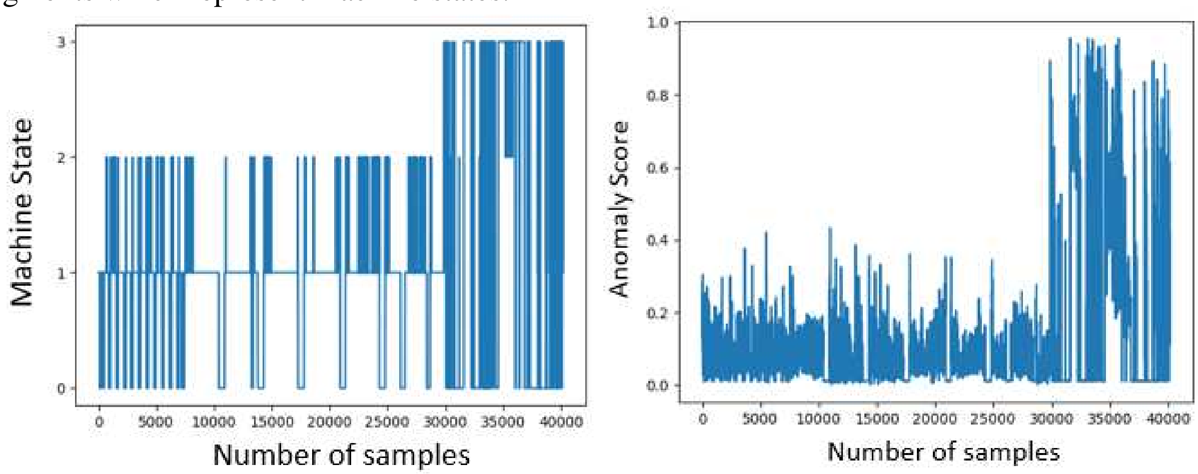
The outputs of each layer are then normalized and added to generate an array of similarity scores using a scoring function. Euclidean distance is used as a score function, which is then translated to loss using a weighted equation. After the similarity scores between each and the number of clusters in each layer is calculated, centers of clusters are updated to minimize the loss. These clusters are initialized by the first few training data with a KNN algorithm before training and continuously updated in the training process. The Adam optimizer was utilized as its faster convergence time than other optimizers [22], with a learning rate of 0.001. In section 5, other model hyperparameters (e.g., learning rate and batch size) are also tested to assess their impact on the model performance.

Finally, a threshold value is selected to identify anomalies as auto selection of a threshold is limited in existing anomaly detection algorithms. Percentiles were utilized for this study as the threshold to identify anomalies. The threshold value needs to be carefully selected as it reflects the prior knowledge of anomalies. Hence, different percentiles were explored and compared with the machine states. 10% percentile of the outcomes were identified as anomalies based on the actual machine states. Lower percentile values may be used for the threshold to ensure capturing anomalies with the compromise of prediction accuracy.

4. Analysis and Discussion

4.1 Classification Results

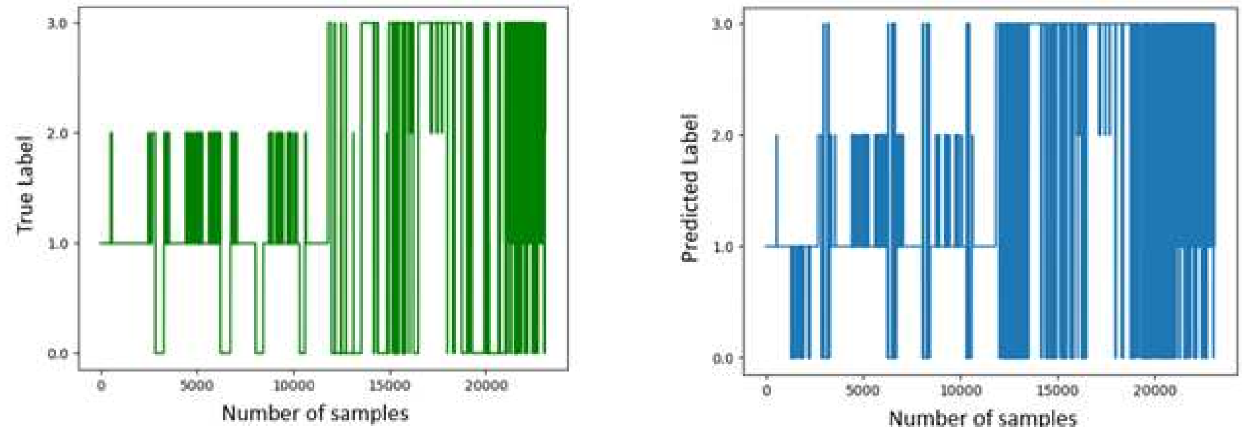
The effectiveness of the proposed network in identifying bearing faults in the air compressor is assessed in this section. The bearing state, which is plotted in blue in Fig. 5 (a), shows the condition of the bearing/machine. A state of 0 indicates the machine is off, 1 indicates a normal condition of the machine, 2 means the machine is nearing failure, and 3 means the machine is running but is in a failure condition. Anomaly scores, which represent how the test data deviates from the normal condition, are shown in Fig. 5 (b). The outcome clearly shows that the anomaly score significantly increased before the bearing failure happened, which is identified near sample number of 32,000. The labels were created in such a way so that the label (true state of the machine) increases as the machine approaches failure. Thresholds are applied to categorize the anomaly scores into four segments which represent machine states.



(a) (b)

Fig. 5 (a) True labels for machine states (stop, normal, near failure, failure), (b) anomaly scores for total dataset

Using threshold values, anomalies are identified and shown in Fig. 6. Anomaly scores are converted into predicted machine states using the threshold values. The threshold values are determined based on the initial percentile distribution of the anomaly scores for each labeled machine state. For the 'stopped' state, which is labeled as 0, the lower and upper thresholds are the 0 and 24.3 percentiles. The 'normal' state, which is the largest portion of the dataset, has its thresholds set at the 24.3 and 84 percentiles. The 'near failure' state, the smallest portion of the dataset, is identified using threshold of 84 and 87.3 percentiles. After applying the thresholds, it turns out that the abnormal operations of the air compressor started around April 2022. The pharmaceutical manufacturing company noticed the bearing failure in early 2023 and halted the machine operation. Repair/maintenance did not immediately occur owing to the unavailability of machine components (e.g., bearings) from the supplier. This productivity loss might have been avoided if the proposed model had been used to identify impending bearing failure.



(a) (b)

Fig. 6 (a) True labels of testing dataset (b) predicted labels for testing dataset after applying thresholds to anomaly scores

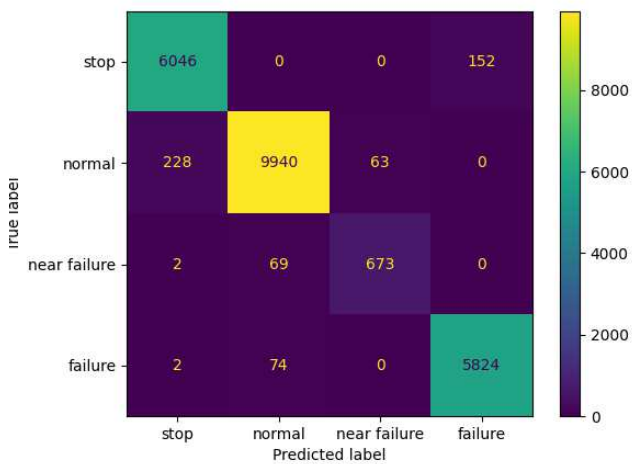


Fig. 7 Confusion matrix of predicted states and true states

The model achieved a high number of true positives for all machine states, as indicated by the large values on the diagonal. This suggests that the model effectively identified instances belonging to these states. However, there were instances of misclassification, especially in normal and failure states, where false positives and false negatives were observed. For example, stop state had 152 instances falsely classified, and failure state had 74 instances falsely classified as normal state.

The confusion matrix reveals a strong performance of the model, with high accuracy in predicting the machine states. However, high prediction accuracy does not necessarily translate to cost-effective business decisions (e.g., reduction in maintenance costs), as false predictions can lead to additional cost for investigating false alarms. That is, there is room for improvement, particularly in reducing misclassifications in certain states (e.g., misclassification of normal state). Examining the specific characteristics of misclassified instances and potentially adjusting the model parameters or features may help enhance the overall performance and accuracy of the proposed model. To improve the model performance in terms of misclassification and overall prediction accuracy, the impact of model hyperparameters is investigated in the following section.

4.2 Impact of model hyperparameters

As noted above, it was suggested that model parameters may be adjusted to enhance the performance of the model. Learning rate (LR) is an important parameter that affects how quickly a model is trained. It is often challenging to determine the appropriate learning rate; a low learning rate increases computational time while a high learning rate may result in a badly trained model. The performance of the proposed network/model was evaluated in this study using three learning rates: 0.001, 0.005, and 0.001.

The amount of data transmitted into a training network before upgrading the model weights is known as batch size. The top-performing hyper-parameters for various batch sizes of 32, 64, and 128 are utilized to evaluate the performance of the proposed network employed in this study. Small values of batch size represent that the model is trained over a small subset of the training data at a particular instance. Such training can result in frequent parameter update resulting in faster convergence. Intrusion of noise and unrepresentative samples are the primary drawbacks of small batch sizes. Conversely, large batch sizes use large subset of data for training. Large batch size training results in a stable parameter update with gradients computed for improved estimation on the overall dataset.

Optimizers play a crucial role in the training phase of machine learning and deep learning models. These algorithms are responsible for iteratively updating the model parameters in order to minimize the loss function and improve the overall performance of the model. The choice of optimizer can significantly influence the convergence speed, stability, and final performance of the trained model. To evaluate the performance of a proposed model, three different optimizers were employed: ADAM, RMSPROP, and SGDM (Stochastic Gradient Descent with Momentum).

Fig. 6 presents a comprehensive analysis of the impact of various model hyperparameters, including the choice of optimizer, LR, and batch size on the overall performance of the model. By comparing the performance metrics obtained with different optimizers, insights can be gained into which optimizer is most effective for the given task.

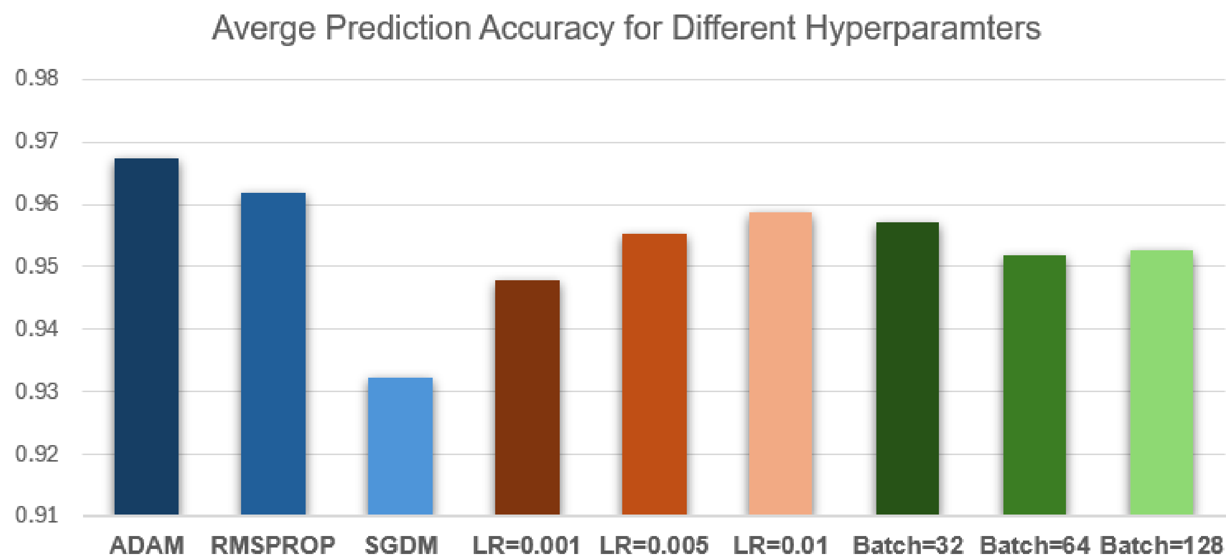


Fig. 8 Impact of hyperparameters on prediction accuracy for the proposed model

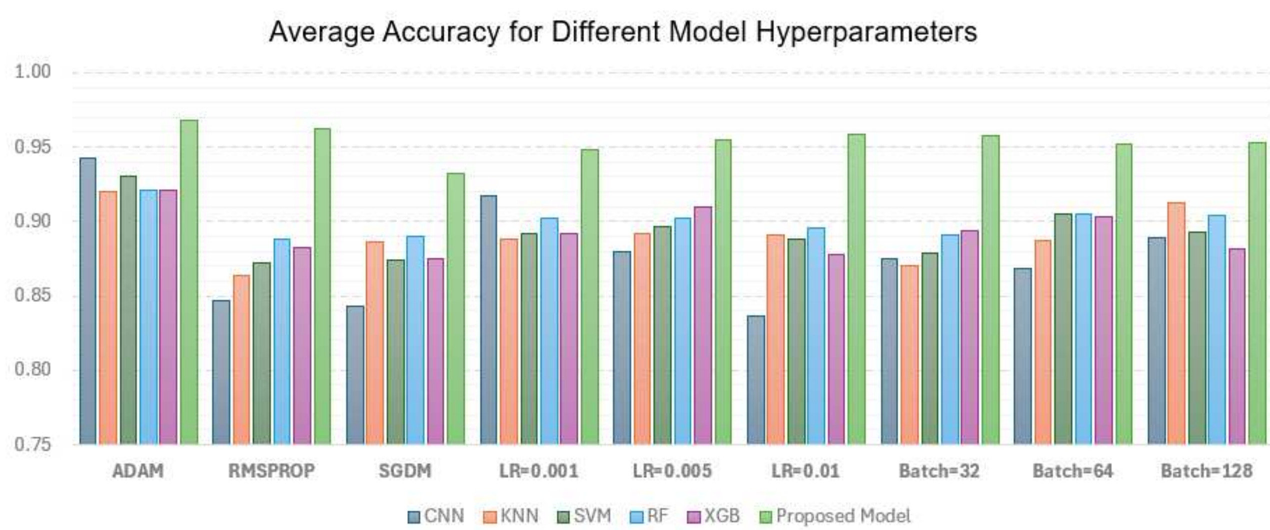


Fig. 9 Comparison of prediction accuracy for different models (CNN, KNN, SVM, RF, XGBoost, Proposed model) and hyperparameters

In order to assess the performance of the proposed model, several baseline models were selected and tested with the same exploration method as the proposed model. CNN, KNN, SVM, RF, and XGBoost were selected, and they achieved 87.7%, 89, 89.2%,90%, and 89.3% of average prediction accuracy, respectively, as shown Fig. 9. The results of experiments show that the proposed model achieved 97.4% of maximum accuracy with 95.3% of overall accuracy, which outperformed the baseline models. The evaluation works with different models and hyperparameters are summarized in Table 1. Overall, the best performance was achieved with the ADAM optimizer and 0.001 of LR. Also, the proposed model achieved the best performance with the ADAM, 0.001 of LR, and 128 of batch size.

Table 1. Summary of model performance

Opt./LR /Batch			Conventional ML Models					Proposed Model
			CNN	**KNN**	**SVM**	**RF**	**XGB**	
ADAM	0.01	32	0.9127	0.9201	0.8991	0.9219	0.9219	0.9612
		64	0.9332	0.9254	0.9054	0.9234	0.9234	0.9685
		128	0.9259	0.9223	0.8976	0.9123	0.9123	0.9721
	0.005	32	0.9154	0.9187	0.9467	0.9287	0.9287	0.9703
		64	0.9332	0.9123	0.9589	0.9165	0.9165	0.9676
		128	0.9658	0.9198	0.9332	0.929	0.929	0.9634
	0.001	32	0.9621	0.9232	0.9156	0.9134	0.9134	0.9657
		64	0.9579	0.9121	**0.9601**	0.9278	0.9278	0.9645
		128	**0.9779**	**0.93**	0.9589	0.9192	0.9192	**0.9744**
RMSPROP	0.01	32	0.8734	0.8165	0.8532	0.8812	0.8981	0.9682
		64	0.7991	0.8312	0.9156	0.8893	0.8937	0.9654
		128	0.7621	0.9234	0.8543	0.8901	0.8155	0.9632
	0.005	32	0.8014	0.8165	0.8532	0.8845	0.8889	0.9691
		64	0.851	0.8312	0.9156	0.8856	0.9312	0.9723
		128	0.9001	0.9234	0.8543	0.8867	0.8365	0.9715
	0.001	32	0.9051	0.8976	0.8227	0.8882	0.9054	0.9692
		64	0.8632	0.8821	0.8812	0.8921	0.8111	0.944
		128	0.8653	0.8498	0.8967	0.8987	**0.9623**	0.934
SGDM	0.01	32	0.7913	0.8356	0.9345	0.8987	0.8332	0.9457
		64	0.7337	0.9234	0.8134	0.9165	0.8998	0.9768
		128	0.7922	0.9167	0.9234	0.8276	0.8011	0.9076
	0.005	32	0.8237	0.8632	0.8398	0.8423	0.9476	0.9245
		64	0.8331	0.9176	0.9276	0.8998	0.8856	0.9121
		128	0.8928	0.9254	0.8365	**0.9476**	0.9249	0.9457
	0.001	32	0.8913	0.8432	0.8412	0.8632	0.8102	0.941
		64	0.9109	0.8523	0.8674	0.8956	0.9398	0.8961
		128	0.9189	0.8998	0.8789	0.9214	0.8333	0.9414
Average Accuracy			**0.8775**	**0.8901**	**0.8920**	**0.9000**	**0.8930**	**0.9539**

5. Conclusion

Traditional maintenance strategies, such as preventive maintenance, may not always be efficient, as they do not take into account the actual condition of the machine. Predictive maintenance approach in air compressor application was investigated and discussed in this work, highlighting the potential benefits of using data-driven techniques such as machine learning and AI in optimizing maintenance schedules. The use of AI-based anomaly detection models can help identify approaching failures before they cause a significant production loss. PdM is capable to increase machine reliability, reduce downtime, and ultimately, lower maintenance costs. A LSTM-RNN based anomaly detection model was developed and utilized to identify bearing failure of an air compressor which is critical asset to maintain the production environment. The results can be further utilized to optimize maintenance schedules, which means that the optimal number of maintenance activities can be conducted without compromising machine reliability, availability, machine and productivity. The model achieved 97.4% of maximum prediction accuracy and 95.3% of overall prediction accuracy. The key takeaways can be summarized as following.

- LSTM-RNN based hierarchical clustering model was proposed to detect abnormal operations of an air compressor. Abnormal operations are caused by bearing failure.
- Tri-axial acceleration data was collected and utilized to train the proposed model.
- Machine condition was labeled as 4 different states: stop, normal, near failure, failure.
- Anomaly scores obtained by the proposed model were converted to predicted machine states.
- The model achieved 97.4% of prediction accuracy, which outperformed baseline models.

Furthermore, the insights garnered from this predictive maintenance strategy can be leveraged to optimize maintenance schedules, ensuring that the optimal number of maintenance activities are conducted without compromising machine reliability, availability, or productivity. By strategically scheduling maintenance tasks based on real-time data and predictive analytics, the risk may be mitigated of unexpected breakdowns and supply chain disruptions without losing the availability/operational efficiency of equipment. As discussed in the previous section, productivity loss may have been avoided by employing predictive maintenance strategies based on the outcomes of the model. Moreover, it is likely that as the machine/bearings began to approach a failure condition, machine performance degraded; again, the model may have helped to avert this situation. This holistic approach to maintenance management represents a paradigm shift towards a more proactive and data-driven maintenance strategy, ultimately driving greater operational stability and cost savings in industrial settings.

Acknowledgement
This work was supported by the Wabash Heartland Innovation Network (WHIN). Any opinions, findings, conclusions, and/or recommendations expressed are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] J. J. Montero Jimenez, S. Schwartz, R. Vingerhoeds, B. Grabot, and M. Salaün, “Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics,” *Journal of Manufacturing Systems*. 2020. doi: 10.1016/j.jmsy.2020.07.008.

[2] M. C. Carnera, “Selection of diagnostic techniques and instrumentation in a predictive maintenance program. A case study,” *Decis Support Syst*, 2005, doi: 10.1016/j.dss.2003.09.003.

[3] K. M. Ngu, N. Philip, and S. Sahlan, “Proactive and predictive maintenance strategies and application for instrumentation & control in oil & gas industry,” *International Journal of Integrated Engineering*, vol. 11, no. 4, 2019.

[4] J. C. A. Jauregui Correa and A. A. Lozano Guzman, “Chapter Eight - Condition monitoring,” in *Mechanical Vibrations and Condition Monitoring*, 2020.

[5] L. Xia, P. Zheng, X. Li, Robert. X. Gao, and L. Wang, “Toward cognitive predictive maintenance: A survey of graph-based approaches,” *J Manuf Syst*, vol. 64, pp. 107–120, 2022, doi: https://doi.org/10.1016/j.jmsy.2022.06.002.

[6] H. Nordal and I. El-Thalji, “Assessing the technical specifications of predictive maintenance: A case study of centrifugal compressor,” *Applied Sciences (Switzerland)*, 2021, doi: 10.3390/app11041527.

[7] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, “Predictive maintenance in the Industry 4.0: A systematic literature review,” *Comput Ind Eng*, 2020, doi: 10.1016/j.cie.2020.106889.

[8] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. G. Jun, and J. W. Sutherland, “Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data,” in *Procedia CIRP*, 2019. doi: 10.1016/j.procir.2018.12.019.

[9] D. Pandya *et al.*, “Increasing production efficiency via compressor failure predictive analytics using machine learning,” in *Proceedings of the Annual Offshore Technology Conference*, 2018. doi: 10.4043/28990-ms.

[10] M. T. Siraj *et al.*, “Selection of Air Compressor for Pharmaceuticals: An Approach with TOPSIS”.

[11] P. Henriquez, J. B. Alonso, M. A. Ferrer, and C. M. Travieso, “Review of automatic fault diagnosis systems using audio and vibration signals,” *IEEE Trans Syst Man Cybern Syst*, vol. 44, no. 5, pp. 642–652, 2013.

[12] B. G. Joung, W. J. Lee, A. Huang, and J. W. Sutherland, “Development and application of a method for real time motor fault detection,” in *Procedia Manufacturing*, Elsevier B.V., Jan. 2020, pp. 94–98. doi: 10.1016/j.promfg.2020.07.002.

[13] W. Shang, P. Zeng, M. Wan, L. Li, and P. An, “Intrusion detection algorithm based on OCSVM in industrial control system,” *Security and Communication Networks*, vol. 9, no. 10, pp. 1040–1049, 2016.

[14] B. Zong *et al.*, “Deep autoencoding gaussian mixture model for unsupervised anomaly detection,” in *International conference on learning representations*, 2018.

[15] M. De Benedetti, F. Leonardi, F. Messina, C. Santoro, and A. Vasilakos, “Anomaly detection and predictive maintenance for photovoltaic systems,” *Neurocomputing*, 2018, doi: 10.1016/j.neucom.2018.05.017.

[16] P. Zheng *et al.*, “Development and application of a method for real time motor fault detection,” *Frontiers of Mechanical Engineering*, vol. 13, pp. 137–150, 2018.
[17] A. Zeid, S. Sundaram, M. Moghaddam, S. Kamarthi, and T. Marion, “Interoperability in smart manufacturing: Research challenges,” *Machines*, vol. 7, no. 2, p. 21, 2019.
[18] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, “Edge computing in IoT-based manufacturing,” *IEEE Communications Magazine*, vol. 56, no. 9, pp. 103–109, 2018.
[19] A. Tsanousa *et al.*, “A review of multisensor data fusion solutions in smart manufacturing: Systems and trends,” *Sensors*, vol. 22, no. 5, p. 1734, 2022.
[20] L. Xiang and A. Hu, “Comparison of Methods for Different Time-frequency Analysis of Vibration Signal.,” *J. Softw.*, vol. 7, no. 1, pp. 68–74, 2012.
[21] L. Shen, Z. Li, and J. T. Kwok, “Timeseries anomaly detection using temporal hierarchical one-class network,” in *Advances in Neural Information Processing Systems*, 2020.
[22] K. K. Chandriah and R. V Naraganahalli, “RNN / LSTM with modified Adam optimizer in deep learning approach for automobile spare parts demand forecasting,” *Multimed Tools Appl*, vol. 80, no. 17, pp. 26145–26159, 2021, doi: 10.1007/s11042-021-10913-0.

Statements and Declarations

1. Funding

This work is supported by the Wabash Heartland Innovative Network (Grant No. 2017 0928).

2. Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

3. Author Contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by ByungGun Joung. The first draft of the manuscript was written by ByungGun Joung and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.