

Predictive maintenance of rotational machinery using deep learning

Mohamed Iyad Ali, Nai Shyan Lai, Raed Abdulla

School of Engineering, Asia Pacific University Technology and Innovation, Kuala Lumpur, Malaysia

Article Info

Article history:

Received Feb 2, 2023

Revised Jul 25, 2023

Accepted Aug 1, 2023

Keywords:

Bluetooth low energy

Deep learning

Long short-term memory

Nnode-red server

Predictive maintenance

ABSTRACT

This paper describes an implementation of a deep learning-based predictive maintenance (PdM) system for industrial rotational machinery, built upon the foundation of a long short-term memory (LSTM) autoencoder and regression analysis. The autoencoder identifies anomalous patterns, while the latter, based on the autoencoder's output, estimates the machine's remaining useful life (RUL). Unlike prior PdM systems dependent on labelled historical data, the developed system doesn't require it as it's based on an unsupervised deep learning model, enhancing its adaptability. The paper also explores a robust condition monitoring system that collects machine operational data, including vibration and current parameters, and transmits them to a database via a Bluetooth low energy (BLE) network. Additionally, the study demonstrates the integration of this PdM system within a web-based framework, promoting its adoption across various industrial settings. Tests confirm the system's ability to accurately identify faults, highlighting its potential to reduce unexpected downtime and enhance machinery reliability.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Raed Abdulla

School of Engineering, Asia Pacific University of Technology and Innovation (APU)

Technology Park Malaysia, Bukit Jalil, Kuala Lumpur-57000, Malaysia

Email: raed@apu.edu.my

1. INTRODUCTION

Rotational machinery is an integral component of a lot of manufacturing companies and other business from various sectors such as construction. Before the emergence of industry 4.0, industrial machines were conventionally maintained by using two main methods, reactive and preventive maintenance. In reactive maintenance, assets fail before being maintained and they are kept running until breakdown and only repaired later. Due to this reason, reactive maintenance causes the highest downtimes and largest drop in industrial productivity. On the other hand, preventive maintenance systems are maintained at fixed intervals to ensure continuous availability. But this method consumes the most resources, has a higher cost as their spare parts are replaced before life cycle is up [1]–[4]. A crucial component of contemporary production systems is maintenance [5]. An advanced maintenance model is necessary due to the complexity of industrial equipment that are made up of several interdependent components [6].

Research conducted by Schleichert [7] analytics institute revealed that on average, predictive maintenance increased the productivity of industrial assets 25%, reduced breakdowns by 70% and lowered maintenance costs by 25%. The most beneficial use of predictive maintenance is maximizing the running time of factories by enabling the repairs to can be carried out before breakdown. The same study also revealed that unplanned downtime is costing industrial manufacturers an estimated \$50 billion each year. The internet of things (IoT) is among the most significant technological advances to have occurred recently. Deep

learning (DL) is a branch of [8]–[10] artificial neural network (ANN) and a subclass of artificial intelligence (AI) that has a big impact on people's lives [11], [12]. Over the last decade, predictive maintenance has become a hot topic of research where many researches have been carried on predictive maintenance using deep learning [13]–[16].

Predictive maintenance systems maximize machine uptime and reduce maintenance expenses. If industrial asset failures are not predicted, downtime and repairs could cost millions. The most common problems that cause downtime in rotational machinery include bearing defects, gear wear, misalignments, unbalance, and insulation damage. These problems usually occur due to environmental factors, human errors, or the self-degradation of these machines. As defects develop in rotational devices such as induction motors and pumps, changes in vibration, motor-current signature, and temperature can be identified. The more complex and sophisticated the machine, the harder it is to monitor gradual changes. Hence data-driven models usually preferred over physical models as they are much more suitable for complex machinery. Deep learning models can be used instead of machine learning models as it eases the model preparation process by automating the feature extraction of data using neural networks. The problem in the matter is to research a deep learning model that can be used for the predictive maintenance of rotational machinery and to monitor the relevant machine parameters data.

Kim *et al.* [17] proposed a deep learning framework that can be used for predictive maintenance without the need of run to failure data history data. The acquisition and pre-processing of raw sensor data for modelling. Subsequently, the model is developed by inputting the pre-processed data into an autoencoder neural network as the training data. Kiangala and Wang [18] suggested an efficient framework that able to detect faults in a conveyor system and significantly cut down on the chance of false faults being detected in a manufacturing conveyor motor. Specifically, a deep learning model was developed in order to determine whether or not the abnormalities found pose a threat to production.

Serradilla *et al.* [19] proposed a predictive maintenance method to detect anomalies and diagnose faults for industrial press machines. Their article intends to address some of the most pressing problems facing industries today, such as improving anomaly detection accuracy, determining the cause of unique failures, and increasing flexibility to shifting environmental conditions. Huang *et al.* [20] suggested a framework based on a machine learning architecture consisting of two stages to accurately identify motor defect modes using only motor vibration signals and no preprocessing. In the majority of anomaly detection approaches, the acquired time domain signals are typically converted to frequency domain.

Hermawan *et al.* [21] proposed an experimental approach for estimating the remaining service life of an aircraft engine. Aircraft engines have minimal fault tolerance; hence a minor issue can lead to devastating results. Therefore, a real-time, accurate engine monitoring system is needed. In order to make an accurate prediction of when an engine would fail, the suggested method combines convolutional neural network (CNN) and long short-term memory (LSTM) neural network models to learn the pattern of the given historical data. Zheng *et al.* [22] suggested a predictive maintenance algorithm based on bi-directional long short term memory (Bi-LSTM) and attention mechanism to increase the reliability and accuracy of rotating machinery failure trend predictions. In their proposed design, a bidirectional LSTM was implemented using PyTorch framework to extract signal data characteristics.

Su *et al.* [23] proposed an unique hybrid deep learning framework for remaining useful life (RUL) prediction that combines a variational autoencoder (VAE) with a time-window-based sequence neural network (twSNN). Low-dimensional and hidden features in the raw sensor data are extracted using VAE in the proposed framework, and a loss function is created to extract valuable data features. Gutierrez *et al.* [24] developed an architecture for a wireless ammeter integrated with open source software in order to enhance maintenance-related decision-making for manufacturing industries. Motor current is chosen as the parameter to be analyzed as anomalies in current is one of the biggest indicators of common induction motor issues such as overload and faulty rotor. In the proposed architecture, the current is measured using a non-evasive split core current transducer connected to an Arduino.

Lygren *et al.* [25] devised an efficient unsupervised solution for predictive maintenance, based on deep learning algorithms, and successfully deployed it to predict and anticipate the failures of an Eni offshore coalescer. In the proposed approach, researchers used a combination of deep learning algorithm based on recurrent neural networks (RNN) autoencoder and a machine learning clustering algorithm. Even though RNN autoencoder is typically considered as an unsupervised algorithm it is also considered as a self-learning method. Cano-Ortega and Sánchez-Sutil [26] successfully developed and implemented a novel long range (LoRa) based induction motor efficiency monitoring system for smart meters. LoRa is the chosen wireless technology for this system due to its large range and low power consumption. The aim of the project is to design a deep learning-based model to predict the failure of a rotational machine and to develop a condition monitoring system for a rotational machine to monitor conditions such as vibration or current.

2. PROPOSED METHOD

2.1. Block diagram

The total working principle of the predictive maintenance system developed for rotational machinery is shown in Figure 1. The first stage of the block diagram is the data acquisition system. The data from the vibration sensor and the current sensor is collected to an Arduino nano 33 IoT board. The collected data is transmitted wireless over Bluetooth low energy (BLE) as a byte stream to a gateway. The program running on the gateway receives the sensor data sent over BLE and Queries the data into an influx dB database hosted on the central server. The second stage of the system block diagram is anomaly detection using an unsupervised deep learning algorithm called LSTM autoencoder, where the output data from this stage then is used for predictive analysis. For the initial step of the anomaly detection stage the raw sensor from the InfluxDB database is extracted, cleaned and exported in the correct format as a comma-separated values (CSV) file by data pipeline script. The exported CSV script is then used to train the autoencoder model, where the trained model is saved and deployed. The trained model is deployed using a flask server, which awaits the live machine sensor data to generate an output.

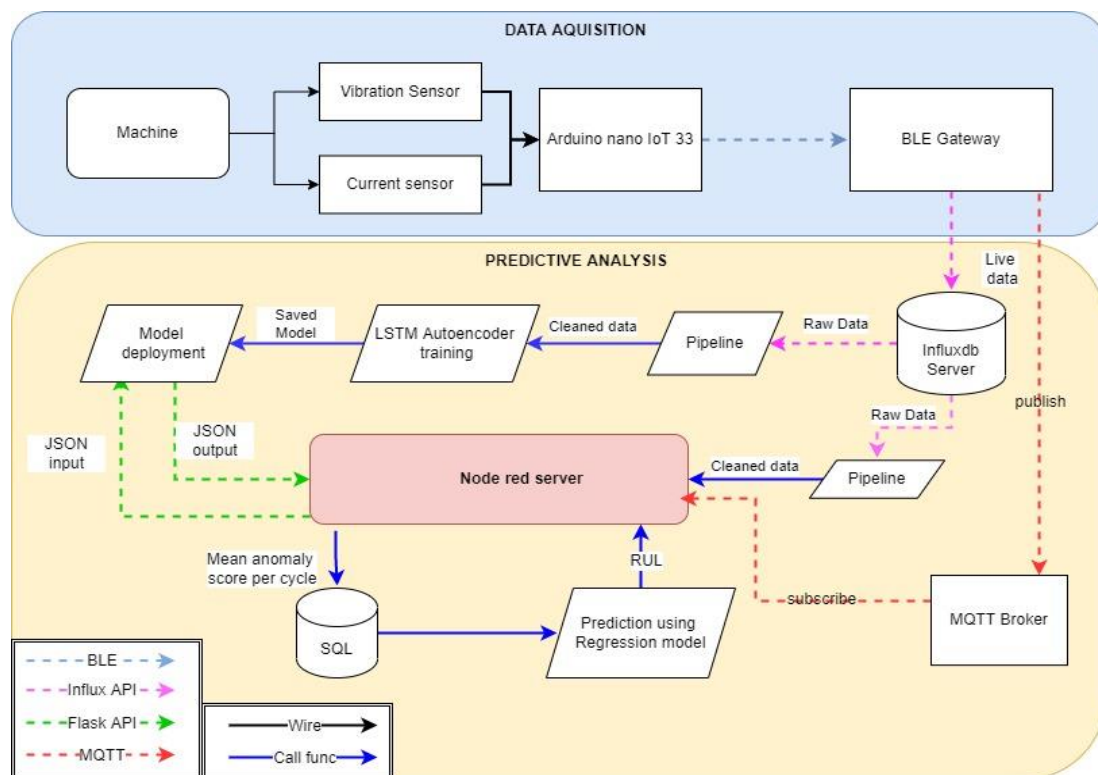


Figure 1. Overall block diagram

A node-red server that serves as the system's backend extracts the current machine data from the InfluxDB server and post the data to the flask server using an application programming interface (API). The deployment script runs the inference on the received CSV data. It produces an output containing the mean anomaly score value and a string which states whether an anomaly has been detected or not, depending on whether the score is greater than the threshold value. The inference output received from the node-red server is then stored along with the cycle number. In the last stage, which is the prediction stage, the mean anomaly score data, along with the cycle number, is trained over a linear regression model. The script calculates the gradient of the plot of anomaly score against the cycle number. Once the gradient is calculated, the remaining number of cycles remaining until the anomaly score reaches the maintenance threshold can be calculated.

2.2. Data acquisition

For the collection of vibration data, KEMET piezoelectric sensor is used, and the current transducer is used to collection of current data. Vibration data is sampled at 20 samples per second with a constant sampling rate to ensure sensitive changes in vibration are recorded. The piezoelectric sensors give the

vibration output in millivolts. The output is taken in millivolts instead of g/m^2 unit in acceleration because the sensitive changes in vibration are detected in millivolts more accurately when compared to acceleration. Furthermore, it has been observed that a loss in accuracy occurs when a 3 significant figure integer is mapped into a float value on Arduino. In addition, since the data is transmitted as a byte stream in BLE, three figure integer in millivolts takes fewer values bytes to transfer the vibration values instead of a float value in acceleration.

A current transducer along with an amplification circuit is used to measure the AC current input to the motor. To reduce the noise in the current readings, a moving average filter is applied as a low-pass finite impulse response (FIR) filter. The library used is called Emon.lib, which is a library written in C to ease the programming for power monitoring. For transmitting the data over BLE, an Arduino library called Arduino BLE is used. The data is converted into 32-bit integers in order to be sent as bytes using BLE. Data transmission in BLE works according to the generic attribute profile (GATT) protocol, where data is either written or read from a Universally Unique Identifier (UUID) which is sent as a byte stream from another BLE device. The working principal of the data acquisition stage can be further explained from Figure 2.

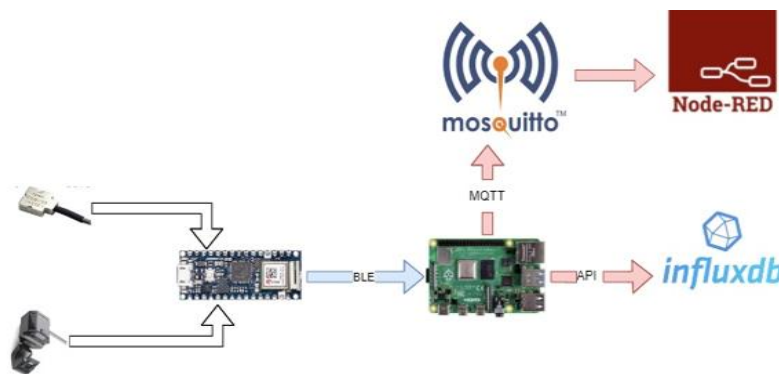


Figure 2. Working principle of the data acquisition system

2.3. Gateway

Raspberry pi is used as a gateway to collect the data from the BLE network and upload it into the dashboard. The flowchart of the gateway program is shown the Figure 3. The tasks which run concurrently are the Bluetooth central function written using python library bleak, which reads BLE byte stream from the sensor node and decodes them into an integer, influx dB application programming interface (API) task, which queries the decoded sensor values into the influx database (DB) and finally, the message queuing telemetry transport (MQTT) publish task which publishes the sensor values to an MQTT broker hence allowing any dashboard to monitor the motor parameters live over the internet.

2.4. Anomaly detection and prediction

The predictive analysis stage is divided into two components, the anomaly score calculation using LSTM autoencoder and the remaining useful life prediction using a regression model. The first step of training the autoencoder is extracting the vibration and current data from the influx dB database and cleaning the data, so it fits the format required by the LSTM autoencoder model. This is using the pipeline script, which extracts the data from the database and produces a CSV file in the correct syntax for training. The data manipulation is done using the Pandas library for columns and NumPy for rows. The CSV file produced by the pipeline is used in the training script of the LSTM autoencoder. CSV data which is put into a data frame is then normalized by passing the data frame through the min-max scaler function. Normalization is done by scaling all the values in the data frame to a value between 0 and 1. Then the timestamp of the data frame is converted to the date time format, and the data frame is reshaped into a 3-dimensional tensor in order to be used in the training process of the LSTM autoencoder. The model parameters are set at 5 layers with two encoder and two decoder layers with an additional repeat vector layer. The activation function of the layers is set to rectified linear unit (ReLU). The flowchart of the working principle of the pipeline and the LSTM autoencoder training script is shown in Figure 4.

Once the training is complete, the model is saved using the python pickle library and deployed in a flask server. Deploying the saved model in a flask server enables the inference to be done online and the results of the inference to be communicated to the backend using an API. After each cycle, the data collected is cleaned and pre-processed by another pipeline script when triggered by a node red server. Then another

node sends the pre-processed data to the deployed model using a representational state transfer (REST) API, as a JavaScript object notation (JSON) file. The deployed inference script converts the received the JSON object into a panda's data frame and run the inference on each column using the saved model from the training. The output of the inference is given as mean a value of the total anomaly score and a string which indicates the anomaly state. The output is posted to the backend node-red server using the Flask API. Once the inference output is received from node red is queries the mean anomaly score and the cycle is a CSV. After the first 10 cycles, the linear regression model is trained and deployed. The prediction script finds the gradient of the plotted anomaly score graph and predicts the number of cycles remaining until the anomaly score reaches the maintenance threshold. The output of the from the code is displayed in the dashboard.

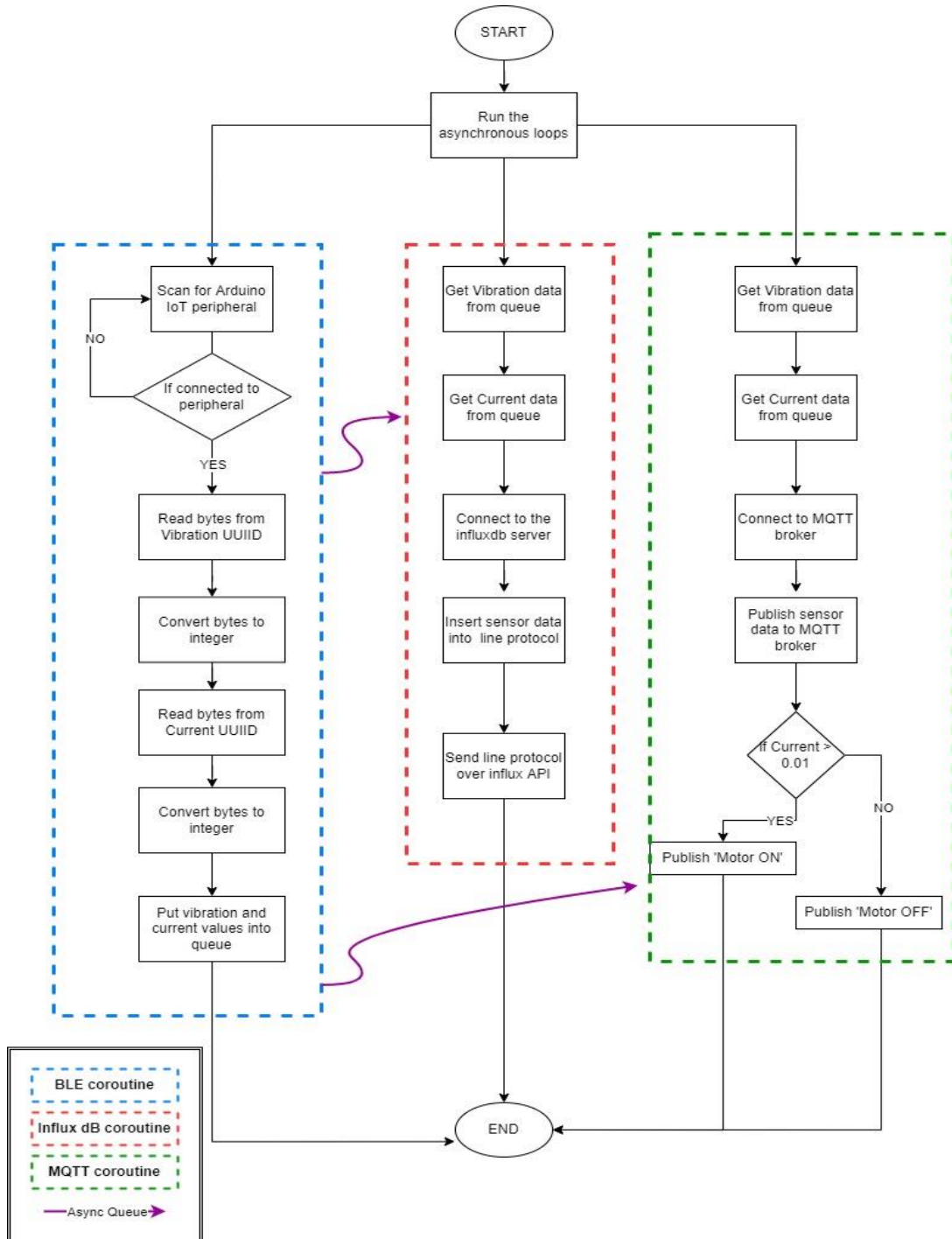


Figure 3. Flowchart of BLE gateway

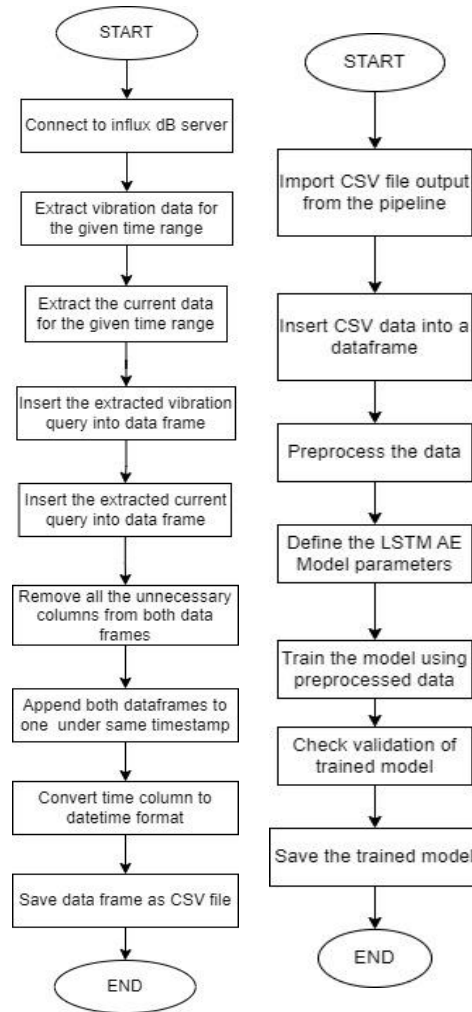


Figure 4. Flowchart of the working principle of the pipeline and the LSTM autoencoder training script

3. HARDWARE AND SOFTWARE RESULTS

3.1. Hardware implementation

The motor is fitted inside an aluminum bracket platform where it is screwed to the aluminum brackets using a 3D printed holder. A variable frequency drive (VFD) is used to control the speed of the direct current (DC) by setting the frequency. Figure 5 shows the full hardware setup made for this project and the 3D printed holder with the sensor placed inside.

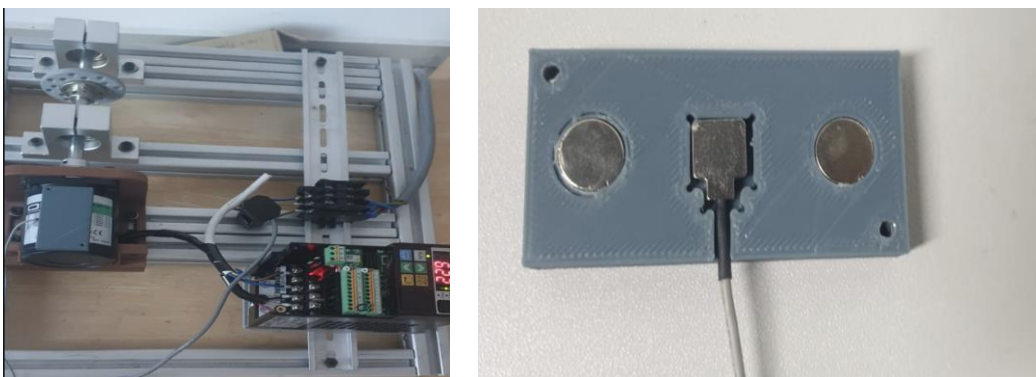


Figure 5. The full hardware setup and 3D printed holder with the sensor placed inside

Figure 6 shows the serial plot of the readings from the current sensor compared with the current sensor readings processed with the moving average filter. Figure 7 shows the data collected by the influx dB database. Implementation of the moving average filter reduced the noise by a considerable amount.

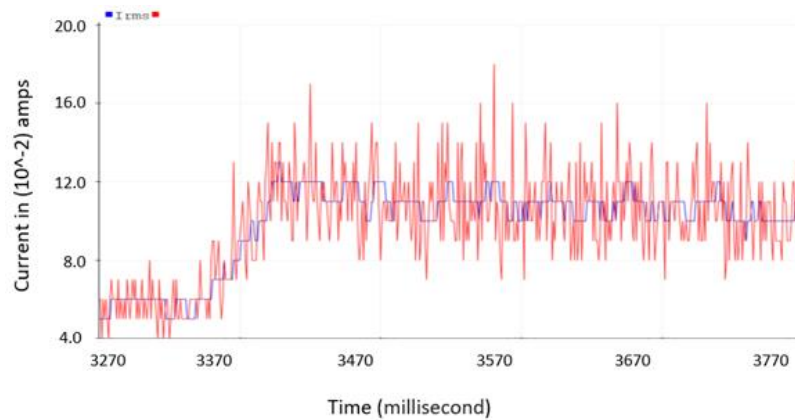


Figure 6. Serial plot of the readings

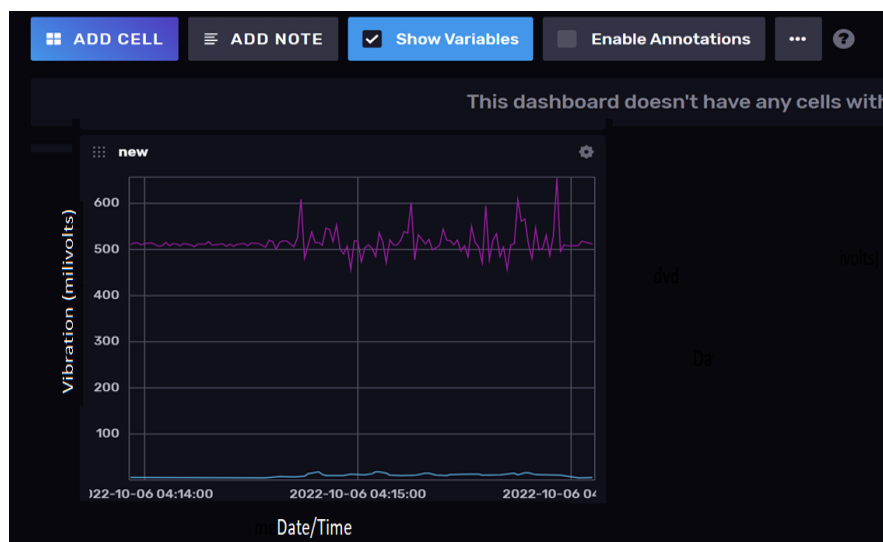


Figure 7. Data collected by the influx dB database

3.2. Range test of the BLE communication

For this test the received signal strength indicator (RSSI) values obtained when the distance between the BLE peripheral (sensor node) and the central (gateway) is recorded and analyzed. RSSI stands for received signal strength and it can be used to test the range of the BLE network. The unit of RSSI is given as dB, as it measures the power of the signal transmitted from its source. Finding the range of BLE network is important because it will determine the maximum distance the gateway devices should be placed from the machine in an industrial setting. To measure the range of the data transmission, the RSSI values are obtained from the Arduino is measured when the distance between the Arduino and the gateway is varied. The experiment is done in a living room with a few obstructions instead of an open area as a possible setup in a factory will also have a few obstructions between the sensor node and the gateway. The graph plotted for the values obtained from the collected samples for the data is shown in Figure 8.

By analyzing the RSSI trend against distance it is observed that the plotted graph represents a logarithmic decline, so the results obtained aligns with the theoretical concepts. As the trend of the curve follows the theory, the experimental setup and the data collection method can be deemed correct. However, during the data collection process, it has been observed that there has been a fluctuation in the RSSI readings

for a same distance. This might be caused due to the sensitivity of the Arduino nano's Wi-Fi chip. According to the results, the furthest distance in which gateway can be placed from the sensor node is 8.5 meters, as that is the maximum range obtained from the experiment.

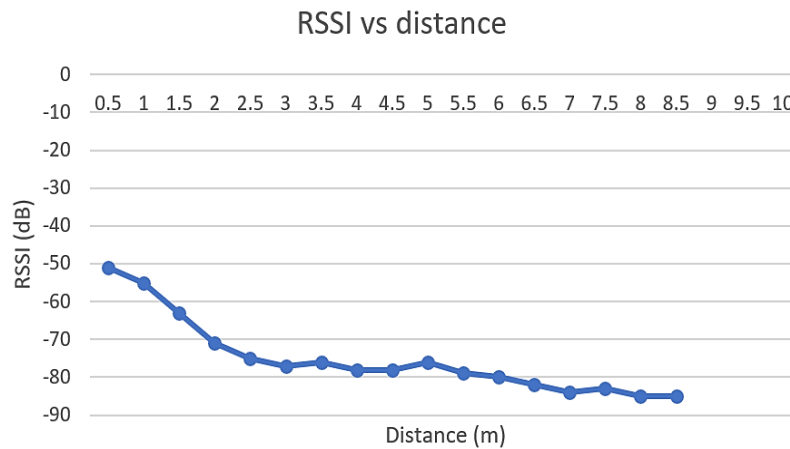


Figure 8. RSSI vs distance results

3.3. Anomaly score validity and prediction test

This test was carried out to analyse the changes in anomaly score and remaining useful life after increasing the vibrations levels from normal. The increase in vibration level was induced by the loosening of some of the screws that were used to attach the motor to the base. The objective of this test is to check the validity of the predictive maintenance model. According to the designed algorithm, any faulty vibration values detected should by the system increase the anomaly score and reduce the remaining useful life as seen in the dashboard. In this experiment, the nuts and bolts used to hold the motor in place is loosened after each run. The loosened nuts and bolts produce bad vibrations which could simulate a faulty motor. The corresponding anomaly score and RUL in cycles seen from the dashboard was noted down. The first reading was with all nuts on the motor mount tightly screwed. After each subsequent run, an extra nut was loosened from the motor mount and the resultant anomaly score and remaining useful life was taken. Table 1 shows the results. From the results obtained, it can be deduced that the developed predictive maintenance system is able to identify faults in a motor and predict the remaining useful life. Anomalies induced during the test such as sudden vibrations due to unstable mounting of the motor resulted in a higher anomaly score. In addition, the remaining useful life in cycles also changes based on the condition of motor data. However, some of the limitations of this experiment include non-linearity of anomaly score and remaining useful life against the machine cycle.

Table 1. Vibration experiment results

Condition	Anomaly score	RUL
No nuts/screws loose	0.08	420
1 nut from the corner loose	0.092	400
2 nuts loose	0.270	350
3 nuts loose	0.328	270

4. CONCLUSION

In conclusion, the project successfully achieved its objective of designing a deep learning model for predicting the failure of rotational machinery. This was accomplished by developing an LSTM autoencoder to calculate anomaly scores and a regression model to estimate the machine's remaining useful life. The unsupervised nature of the deep learning model enhances its applicability in the industry. The trained model was deployed on a Flask server, making the inference process more practical. Node-RED was used to route the output of the anomaly detection model to the regression model for output prediction. The second part of the project involved the development of a wireless condition monitoring system. This system collects machine parameters such as vibration and current using an Arduino Nano and transmits them to a gateway

using the BLE protocol. A Python script was written to collect data from the BLE byte stream, query it into an InfluxDB database, and publish the data to an MQTT broker for live monitoring via an external dashboard tool. Data visualization was achieved using Node-RED and InfluxDB dashboards. The project faced limitations due to the absence of past motor failure data. To accurately validate the developed predictive maintenance system, the motor would need to be run continuously until it exhibits faulty behavior, a task for future research. The system's web-based design, which allows data transfer between individual components through APIs and message queues, opens up the possibility for deployment on a cloud environment such as AWS. This would significantly improve scalability, enabling the system to be used across multiple machines, thus presenting a promising direction for future enhancements.




REFERENCES

- [1] N. Jamil, M. F. Hassan, S. K. Lim, and A. R. Yusoff, "Predictive maintenance for rotating machinery by using vibration analysis," *Journal of Mechanical Engineering and Sciences*, vol. 15, no. 3, pp. 8289–8299, Sep. 2021, doi: 10.15282/jmes.15.3.2021.07.0651.
- [2] H. M. Hashemian and W. C. Bean, "State-of-the-art predictive maintenance techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 10, pp. 3480–3492, Oct. 2011, doi: 10.1109/TIM.2009.2036347.
- [3] H. M. Hashemian, "Wireless sensors for predictive maintenance of rotating equipment in research reactors," *Annals of Nuclear Energy*, vol. 38, no. 2–3, pp. 665–680, Feb. 2011, doi: 10.1016/j.anucene.2010.09.012.
- [4] T. D. Popescu, D. Aiordachioaie, and A. Culea-Florescu, "Basic tools for vibration analysis with applications to predictive maintenance of rotating machines: an overview," *The International Journal of Advanced Manufacturing Technology*, vol. 118, no. 9–10, pp. 2883–2899, Feb. 2022, doi: 10.1007/s00170-021-07703-1.
- [5] C. Chen, Y. Liu, X. Sun, C. Di Cairano-Gilfedder, and S. Titmus, "Automobile maintenance prediction using deep learning with GIS data," *Procedia CIRP*, vol. 81, pp. 447–452, 2019, doi: 10.1016/j.procir.2019.03.077.
- [6] A. Van Horenbeek and L. Pintelon, "A dynamic predictive maintenance policy for complex multi-component systems," *Reliability Engineering and System Safety*, vol. 120, pp. 39–50, Dec. 2013, doi: 10.1016/j.res.2013.02.029.
- [7] O. P. Schleichert, B. Bringmann, H. Kremer, S. Zablotskiy, and D. Kopfer, "Predictive maintenance: Taking pro-active measures based on advanced data analytics to predict and avoid machine failure." 2017.
- [8] H. A. Khan, R. Abdulla, S. K. Selvaperumal, and A. Bathich, "IoT based on secure personal healthcare using RFID technology and steganography," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, p. 3300, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3300-3309.
- [9] L. M. Thakur, R. Abdulla, S. K. Selvaperumal, and C. Nataraj, "Pesticide sprayer for agricultural purpose based on IoT technology," in *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, Oct. 2022, pp. 414–421, doi: 10.1109/ICOSEC54921.2022.9952066.
- [10] R. Lakshmanan, M. Djama, S. Perumal, and R. Abdulla, "Automated smart hydroponics system using internet of things," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 6389–6398, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6389-6398.
- [11] K. Wang and Y. Wang, "How AI affects the future predictive maintenance: a primer of deep learning," in *Lecture Notes in Electrical Engineering*, Springer Singapore, 2018, pp. 1–9.
- [12] M. Al-Shabi and A. Abuhamdah, "Using deep learning to detecting abnormal behavior in internet of things," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 2108–2120, Apr. 2022, doi: 10.11591/ijece.v12i2.pp2108-2120.
- [13] A. Nasser and H. AL-Khazraji, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 721–730, Feb. 2022, doi: 10.11591/ijece.v12i1.pp721-730.
- [14] J. Dalzochio *et al.*, "Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges," *Computers in Industry*, vol. 123, Dec. 2020, doi: 10.1016/j.compind.2020.103298.
- [15] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, Nov. 2019, doi: 10.1016/j.cie.2019.106024.
- [16] T. Caroff, S. Brulais, A. Faucon, A. Boness, A. S. Arrizabalaga, and J. Ellinger, "Ultra low power wireless multi-sensor platform dedicated to machine tool condition monitoring," *Procedia Manufacturing*, vol. 51, pp. 296–301, 2020, doi: 10.1016/j.promfg.2020.10.042.
- [17] D. Kim, S. Lee, and D. Kim, "An applicable predictive maintenance framework for the absence of run-to-failure data," *Applied Sciences*, vol. 11, no. 11, Jun. 2021, doi: 10.3390/app11115180.
- [18] K. S. Kiangala and Z. Wang, "An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment," *IEEE Access*, vol. 8, pp. 121033–121049, 2020, doi: 10.1109/ACCESS.2020.3006788.
- [19] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects," *Applied Intelligence*, vol. 52, no. 10, pp. 10934–10964, Aug. 2022, doi: 10.1007/s10489-021-03004-y.
- [20] Y. Huang, C.-H. Chen, and C.-J. Huang, "Motor fault detection and feature extraction using RNN-based variational autoencoder," *IEEE Access*, vol. 7, pp. 139086–139096, 2019, doi: 10.1109/ACCESS.2019.2940769.
- [21] A. P. Hermawan, D.-S. Kim, and J.-M. Lee, "Predictive maintenance of aircraft engine using deep learning technique," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2020, pp. 1296–1298, doi: 10.1109/ICTC49870.2020.9289466.
- [22] L. Zheng *et al.*, "Rotating machinery fault prediction method based on Bi-LSTM and attention mechanism," in *2019 IEEE International Conference on Energy Internet (ICEI)*, May 2019, pp. 53–58, doi: 10.1109/ICEI.2019.00016.
- [23] C. Su, L. Li, and Z. Wen, "Remaining useful life prediction via a variational autoencoder and a time-window-based sequence neural network," *Quality and Reliability Engineering International*, vol. 36, no. 5, pp. 1639–1656, Jul. 2020, doi: 10.1002/qre.2651.




- [24] S. Gutierrez, L. Islas, F. M. Ibanez, M. Cardona, J. Calzada, and V. K. Solanki, "Wireless ammeter based on zigbee for continuous monitoring of induction motors," in *2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)*, Nov. 2019, pp. 1–6, doi: 10.1109/CONCAPANXXXIX47272.2019.8977048.
- [25] S. Lygren, M. Piantanida, and A. Amendola, "Unsupervised, deep learning-based detection of failures in industrial equipments: the future of predictive maintenance," *Abu Dhabi International Petroleum Exhibition & Conference*, Nov. 2019, doi: 10.2118/197629-MS.
- [26] A. Cano-Ortega and F. Sánchez-Sutil, "Monitoring of the efficiency and conditions of induction motor operations by smart meter prototype based on a LoRa wireless network," *Electronics*, vol. 8, no. 9, Sep. 2019, doi: 10.3390/electronics8091040.

BIOGRAPHIES OF AUTHORS






Mohamed Iyad Ali    received the B.Eng. degree in telecommunications engineering from Asia Pacific University of Technology and Innovation (APU), Malaysia, in 2022. He served as a committee member for APU ImechE student section and as well as a member APU IEM student section. His research interests include applications of artificial intelligence in industry 4.0, internet of things (IoT) and wireless communications. He can be contacted at email: iyadhu213@gmail.com.



Nai Shyan Lai    is currently an associate professor at School of Engineering, Asia Pacific University of Technology and Innovation, working in the area of micro-electronics and nano-electronics. He received his B. Eng. (Hons) in electrical engineering (2007) and Ph.D. in the field of silicon-based nano-electronics and radiation detectors (2012) from The University of New South Wales. His main research interests include semiconductor micro-fabrication and nano-fabrication, quantum dot devices, cryogenic temperature measurements, single electron transistors, quantum computation and silicon microdosimeters. Since 2009, he has published most of his research work in IEEE Transactions, American Institute of Physics, American Physical Society, and Nature Group journals. He can be contacted at email: nai.shyan@apu.edu.my.



Raed Abdulla    is currently an associate professor at School of Engineering, Asia Pacific University of Technology and Innovation. He received his B.Sc. in Electrical Engineering from AL-Mustansiria University, Baghdad, Iraq in 1997, and the M.Sc. degree in Electronic Systems Design Engineering from Universiti Sains Malaysia (USM), Malaysia in 2008 and he obtained his Ph.D. in wireless and mobile systems from USM in 2012. His main research interests include wireless system design, radio frequency identification (RFID), wireless sensor network (WSN) and internet of anything (IoA), machine learning applications for wireless communication systems, and green communications. He can be contacted at email: raed@apu.edu.my.