Implementation of a Predictive Maintenance System using Unsupervised Anomaly Detection

Oscar S. Kremer * **Mauro A. B. Cunha** ** **Paulo H. A. Silva** ***

* *Electrical Engineering, Sul-riograndense Federal Institute, Pelotas, (e-mail: oscar.s.kremer@hotmail.com).*
** *Electrical Engineering, Sul-riograndense Federal Institute, Pelotas, (e-mail: maurocunha@ifsul.edu.br)*
*** *Technology for Internet Systems, Sul-riograndense Federal Institute, Pelotas, (e-mail: paulosilva@ifsul.edu.br)*

Abstract: This work presents the implementation and architecture of a predictive maintenance system using unsupervised learning, aiming to provide an approach where its benefits are returned to the users faster than traditional system found in the literature. Equipment faults and maintenance affect directly efficiency of industrial plants. Maintenance management can be optimized by modeling and predicting problems. The implementation of a system able to model problems or anomalies and to inform operators and supervisors in case of alarms or notifications is intrinsically connected with the fourth industrial revolution. Although the architecture proposed here can be generalized to many different devices, a MPU6050 for sensing and ESP32 as the middleware were used to implement the proposed concept. Three different methods for anomaly detection were implemented and deployed to Google Cloud, using Compute Engine service. The developed web-server that provides a dashboard to visualize time-series of the sensed physical magnitudes and historical data about all anomalies detected is presented. Therefore, resulting in a platform with intelligence to detect and report problems and abnormalities to employees of industrial plants.

Keywords: Predictive Maintenance, Anomaly Detection, Industry 4.0, IoT, Cloud Computing.

1. INTRODUCTION

Ever since the beginning of industrialization, technological leaps have led to paradigm shifts. Some of the biggest paradigm shifts in humanity are named, today, as industrial revolutions. Historically, one can refer to three industrial revolutions. Firstly in the field of mechanization and introduction of steam machines, secondly in intensive use of electrical energy and lastly the digitalization and information era (Lasi et al., 2014). Today, a new paradigm shift is happening, based on principles such as IoT (Internet of Things), cloud computing, big data & analytics, smart devices, among others. It has had an enormous impact on our society, and has been entitled as the fourth industrial revolution or Industry 4.0 (Schwab, 2017). To reach a better understanding of the scale of these impact, according to the Wall Street Journal artificial intelligence has the potential to increase productivity near 1.2% between now and 2030. Such increase is bigger than the ones caused by introduction of steam machines, robotics and even the internet, which are equal to 0.3%, 0.4% and 0.6%, respectively (WSJ, 2018).

Among others, some of the main characteristics of industry 4.0 are: flexibility, individualization on demand, decentralization, resource efficiency and short development periods (Lasi et al., 2014; Rüßmann et al., 2015). Efficiency has an important role in this revolution and during an industrial process may refer not only to material, but also time and energy, which are relevant to production costs. Therefore, the role of equipment maintenance in quality control and cost reductions is more evident than ever (Lee et al., 2006). Depending on the type of industry the costs of maintenance can represent between 15% and 70% of production costs (Mobley, 2002), besides that, due to uncertainties and management/planning inefficiencies about 33% of this cost is wasted (Mobley, 2002). This great economic impact makes maintenance an important bottleneck in the optimization of an industrial process.

Maintenance management can be divided into three different groups: corrective, preventive and predictive (Mobley, 2002). Corrective maintenance (also known as reactive maintenance) or run-to-failure maintenance is a management approach which machines are fixed only after problems occur. Preventive maintenance is a time-driven management technique which focus on the machine mean-time-to-failure statistics to execute certain routines so that it will be able to run safely for a new period of time. Lastly, predictive maintenance (PdM), the newest management paradigm, is an approach that strives to model degradation and detect anomalies based on real-time measures, not having pre-scheduled routines, allowing the equipment to be explored to its maximum before it reaches failure (Mobley, 2002; Velmurugan and Dhingra, 2015).

Different architectures have been proposed in the literature for PdM systems. In Cachada et al. (2018), a PdM system is presented, using cloud data storage, expert systems for anomaly detection and machine learning algorithms which data inserted from HMI's (Human Machine Interface) installed in the industrial plant are used as input. Other architecture for PdM is proposed in Chiu et al. (2017), using a cyber-physical agent and AMCoT (Advanced Manufacturing Cloud of Things), a platform proposed in Cheng et al. (2016) as way to supply the minimum maintenance requisites for industry 4.0 . However, some difficulties arises when implementing PdM, since it involves a change in corporate attitude and culture, may have high installation cost and, depending on the system and algorithms used, could take a long time to reach the full benefits of its implementation.

In the present work an architecture for PdM is proposed, using IoT, cloud computing, big data, smart sensors and artificial intelligence. A proof of concept is used to show its implementation. The work aims to implement an architecture capable of overcoming some difficulties that reduce PdM acceptance. Most architectures use supervised machine learning (Cachada et al., 2018; Chiu et al., 2017; Cheng et al., 2016), being limited to specific types of equipment or the presence of datasets for classification. The solution of this problem depends on the application of unsupervised learning, removing the need for labeled data and also increasing the speed for implementation and installation. This article is organized as follows. Section 2 approaches a theoretical overview of PdM and some technological pillars of the fourth industrial revolution. In section 3, the proposed architecture is described. An illustrative example is implemented and presented in section 4, including hardware specification, libraries and algorithms. The results of the implemented concept are presented in section 5. Final considerations are presented in section 6.

2. PREDICTIVE MAINTENANCE SYSTEMS

Among the categories of maintenance policies, the predictive was the last to be proposed and has been accepted and adopted in different areas, specially in dangerous scenarios, e.g., transmissions lines, nuclear plants and transport systems. This policy involves the prediction of failures/faults in the system or equipment based on the processing of current and/or past values (Mobley, 2002). PdM can be used for both diagnostic and prognostic, allowing a better interpretation of problems and measures. In its older version, already proposed in 1940, professionals would use their senses to predict problems. The main change of paradigm considering the modern PdM is that human senses have been replaced by sensors and the evaluation that would be done considering individual elements now can be done considering the entire system (Selcuk, 2017).

To implement systems as the proposed in Cachada et al. (2018); Chiu et al. (2017); Cheng et al. (2016) one needs to know which variables are important and relevant for the process. Measurements of parameters such as humidity, temperature, vibration, electric current, voltage and impedance allow analysis of a variety of applications, as engines, turbines and hydraulic systems (Hashemian, 2010). The reliability of sensors used is of great importance. Advances in microelectronic resulted in electronic devices capable of executing pre-processing techniques, multi-sensing, self-calibration and communication. The combination of a basic sensing element, embedded intelligence and processing capabilities defines a smart sensor (Giachino, 1986; Hunter et al., 2010).

In addition to smart sensors, communication is an important characteristic to fully implement a PdM architecture. IoT is a communication network for connecting every physical object in the real world (Perera et al., 2013), an essential element to cyber-physical systems and, consequently, to industry 4.0 (Cheng et al., 2016). This network has been considered the promising technology for IT infrastructure, allowing the connection of analog and digital hardware, middleware and advanced software. Another promising technology is cloud computing, an emerging trend well suitable for scenarios where computational power, accessibility, agility and scalability are necessary. The extension of cloud computing to industry results in the concept of cloud-based manufacturing, a vital element for IoT and cyber-physical systems (Huang et al., 2015). According to Mell et al. (2011), cloud computing must present the following key characteristics:

- On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops and workstations).
- Resource pooling: The provider computing resources are pooled to serve multiple consumers by using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state or data-center). Examples of resources include storage, processing, memory and network bandwidth.
- Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer.

Lastly but extremely important, a PdM system must have an internal mechanism capable of providing prognostics or diagnostics to detect failures and abnormalities (Cachada et al., 2018). Anomaly detection is an important task and can be applied to different areas, being extremely consonant with PdM purpose, since its goal is to detect

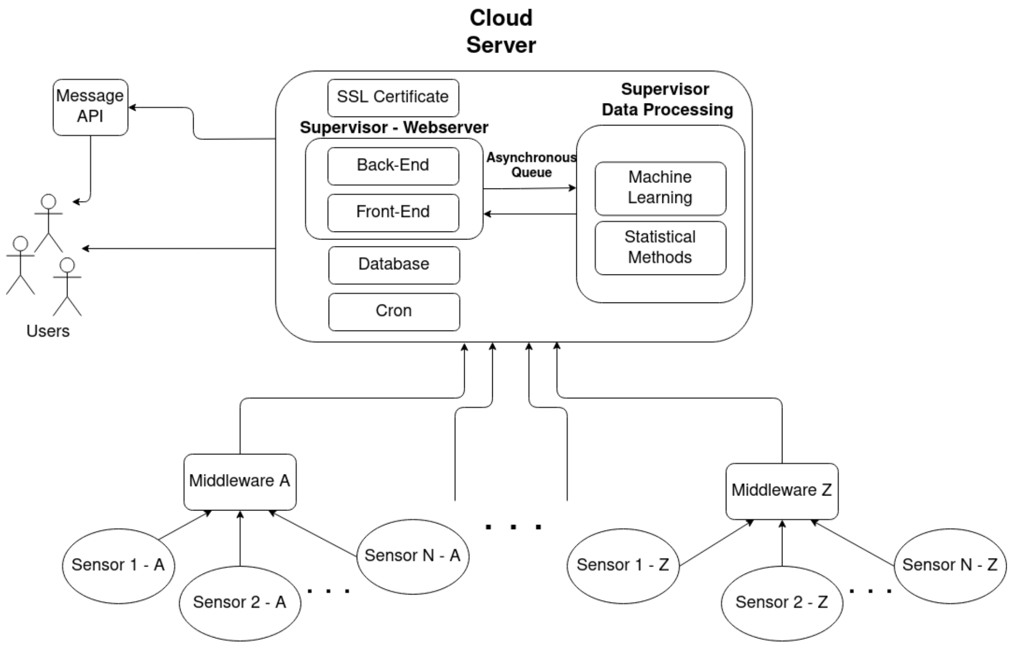


Figure 1. Architecture diagram

patterns in data that does not obey a pre-determined behavior (Chandola et al., 2009). A great challenge when applying anomaly detection is how to define patterns for non-anomaly behavior without removing or camouflaging anomalies created by suspicious agents, knowing that the concept of anomaly can change according to application or context. Another problem is related with obtaining labeled datasets in case one uses supervised algorithms (Chandola et al., 2009), and since sensors measure variables as functions of time, it is important to use algorithms suitable for sequential data (Teng, 2010). Models for anomaly detection in time series include supervised approaches such as neural networks, support vector machines, ensemble learning and unsupervised as statistical models, clusterization and deep-learning models (Ahmad et al., 2017; Munir et al., 2018; Amruthnath and Gupta, 2018).

3. PROPOSED ARCHITECTURE

Technologies such as IoT, cloud computing, smart sensors, big data & analytics are essentials for PdM. Even thought the benefits of using a predictive policy are vast, the high cost and time it takes to fully reach its results collaborate for some industries to choose preventive or even corrective methods. The architecture presented in this work focus on using only unsupervised algorithms for analyzing collected information from installed sensors. A diagram of the system is shown in Figure 1.

The architecture considers an arbitrary set of sensors connected to an arbitrary set of middlewares. The sensors can be both traditional analog systems or more sophisticated smart sensors. The presence of middleware is essential for connection, since in this scenario the middleware act as a centralizer and as bridge to allow communication between sensors and a cloud server. The communication between server and middleware can be done using different approaches, including HTTP, MQQT or even JSON-RPC if communication is made via web-socket.

The cloud server is the main element of the system and it consists of seven sub-elements:

- Database: Responsible for storing data from different entities and for creating relations between elements of different tables.
- Back-End: Bridge responsible for connecting the interface with the data collected from the sensors and stored in the database. In this architecture the back-end is also responsible for sending new measures to algorithms for classification.
- Front-End: Responsible for the visualization of the information. It is of great importance to build interfaces that are responsive and reactive, guaranteeing a better user experience.
- SSL certificate: Certification to allow the server to run with HTTPS protocol instead of HTTP. Such certificate protect users and the server from malicious agents.
- Cron: Daemon that executes scheduled commands, this commands include messages through email, SMS or via external API's for notification and report.
- Machine Learning: Robust and complex algorithms used to detect patterns in data that could be classified as anomalies.
- Statistical Methods: Classical approaches with lower computational cost, to provide information about the behavior of measured values through time.

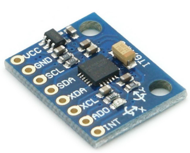
The back-end, front-end and algorithms must run continuously, needing to be constantly active. A task control system must be installed to control theses process. Even thought algorithms run independent from the web application, a communication between them must exist. To detect if an anomaly has happened, the measures received in the back-end must be sent to algorithms for processing. An asynchronous service is used to avoid bottlenecks in the computational power of the server, allowing control of available processing resources for algorithms, preventing the interface from being compromised. The page and information can be accessed from the static IP of the server.

4. ILLUSTRATIVE EXAMPLE

An example server was developed to demonstrate the proposed system structure. As presented by Hashemian (2010), vibration and temperature are some of the most important variables to be measured for fault detection in an industrial environment. In this work, the sensor MPU6050 was chosen. MPU6050 is a microelectromechanical system used as accelerometer and gyroscope with 6 degree of freedom. The sensor also includes temperature measurement, I2C and SPI interface for communication and embedded pre-processing techniques for signal conditioning (INVENSENSE, 2013), such characteristics define MPU6050 as a smart sensor. To guarantee connection to the internet an ESP32 Dev Toolkit (Espressif, 2020) was used as middleware and connected to MPU6050 via I2C.

ESP32 is a SoC (System on Chip) widely used in IoT applications (Maier et al., 2017), mainly because of its integrated antenna and oscillators, combined with low cost and power consumption. More complex devices as Raspberry Pi have a bigger computational power and can also be used for IoT, but since this devices contain embedded operational systems, real-time requisites are normally not attended. In Fig. 2a and 2b ESP32 and MPU650 are shown, respectively. The middleware send HTTP request containing the information captured from MPU6050 encapsulated in a JSON, the requests frequency was set to 0.1 Hz, providing enough time for the algorithms in the server to classify each reading. To compensate for the low sampling frequency rate used, internally the middleware executes 100 readings before sending the information to server, including in each request the median value and also the first and ninth decile of each magnitude, creating an error band analogous to the one proposed in Taylor and Letham (2018).

(a) ESP32 Dev-Toolkit



(b) MPU 6050

Figure 2. Electronic devices used as sensor and middleware

To deploy such application an IaaS (Infrastructure as a Service) is necessary, since other cloud services rarely allow the level of customization needed. In the present work Compute Engine, an IaaS in Google Cloud platform, was used. A machine was instantiated and accessed via SSH for initial configuration. In the server the following software were installed: supervisorctl 4.2.2, nginx 1.21.0 and celery 5.10.0. Supervisorctl is a client/server system that allows its users to monitor and control a number of processes on UNIX-like operating systems. Nginx is a HTTP and reverse proxy server, known for high performance, stability, simple configuration and low resource consumption. Celery is a simple, flexible and reliable distributed system to process vast amounts of messages, being a task queue with focus on real-time processing, while also supporting task scheduling. Celery is specially important due to the high computational cost of some algorithms used for anomaly detection.

Two classic and one deep learning based approach were chosen as unsupervised anomaly detection algorithms. The classic methods include outlier detection and frequency domain analysis. Outlier detection is a well-known statistic method to detect measures that do not follow the normal distribution in the dataset, classifying as anomalies those points which magnitude is greater than the third quartile plus inter-quartile distance or smaller than the first quartile minus the inter-quartile distance (Ahmad et al., 2017). Frequency analysis on other hand aims to detect signal patterns that not necessarily causes outliers but are related to non desired frequencies, as it occurs for high frequency noise. To implement frequency analysis FFT algorithm was used.

The deep learning based method implemented was DeepAnT (Munir et al., 2018). DeepAnT consists of a CNN (convolutional neural network) used to predict time-series. The CNN creates predictions for each point of the signal considering the past-values. The real values are compared with the estimations, and since the CNN creates a prediction based on the training with past measures, it is assumable DeepAnT output will follow the correct pattern. The distance between each prediction and real measure is computed to create a score. A pre-defined threshold defines which predictions are or are not considered anomalies. Others algorithms can be used in the place of CNN, for example, LSTM, SARIMA and others traditional forecasting methods. CNN was chosen because it is a topology of neural network model extremely popular in the literature.

Different from the classic approaches DeepAnT is not applied to only one measure, its input is a matrix NXM, where N is the number of samples stored and M is the number of variables, so that the prediction output would be a M sized array. This work used 21 different variables as input for DeepAnT, representing acceleration in three axis, rotation also in three axis, temperature and uncertainty bands of each one of this measures. In the end, DeepAnT is capable of detecting anomalies present in each variable and also patterns that when examined individually do not appear to be anomaly, however, combined can be undesired, consonant with the intentions of modern PdM (Selcuk, 2017).

The web-server was developed using the programming language python 3.7 and libraries: dash 1.9.0, plotly 4.5.1, flask 1.1.2 and sqlalchemy 1.3.22. Dash is a python library which allows the development of reactive applications and Flask is a more traditional framework, that can be used to create user authentication, pages for login, registration and user settings update. Sqlalchemy is an object-relational mapper (ORM) used to implement database tables. The users registered in the system can receive notifications of anomalies via WhatsApp in case of any problems, for such feature Twilio (2021) API was used to create a maintenance assistant. Others libraries were also used as torch 1.7.1, scikit-learning 0.20.1, scipy 1.5.4, numpy 1.15.4 and pandas 0.24.0. The project code is open-source and is available on `https://github.com/oscarkremer/predictive-maintenance-web`.

5. RESULTS

The developed interface is shown in Fig. 3 for mobile and desktop platform. In the interface it is possible to visualize a real-time graphic containing the acceleration, rotation and temperature in the last 30 minutes. Error band can be deactivated and time interval of the chart can be changed. Another chart is added for historical data, if one wants to analyze all the stored information. The dashboard also shows information about the anomalies that have occurred and on which physical magnitude has happened. In the case of DeepAnT, the anomaly is not related to only one variable, and for its implementation some constraints were considered.

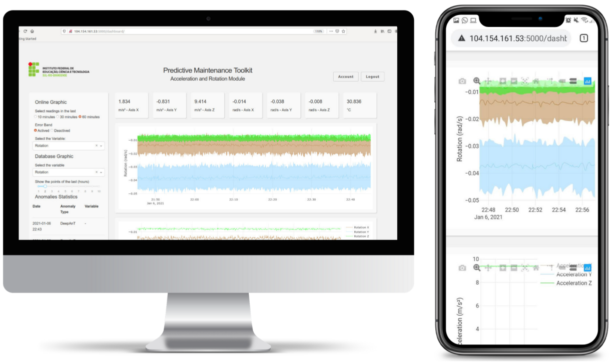


Figure 3. User interface for mobile and desktop

Firstly, to avoid a bad performance due to the lack of data, which could cause the algorithm to return too many false positive anomalies, a minimum number of 500 points was defined. An example output of DeepAnT trained with 200 epochs and the minimum amount of samples for 40 minutes of reading can be seen in Fig. 4. The threshold defined to classify each point as anomaly or not was 3σ, where σ is the standard deviation of stored information.

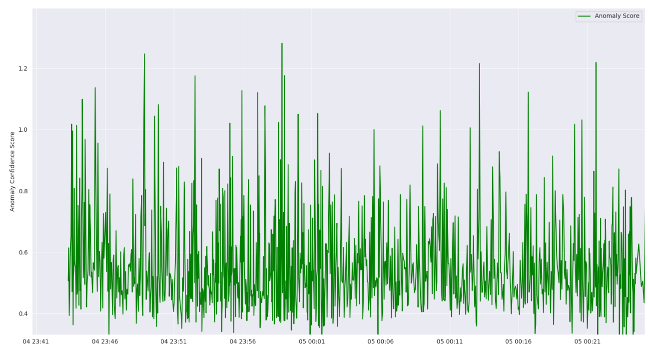


Figure 4. Example of anomaly score through time

Even though there are peaks in Fig. 4, its possible to see that the last measure contain a reduced score, consequently not being classified as an anomaly. The statistical distribution of the anomaly score can be seen in Fig. 5, where one can verify that the scores naturally follow a normal distribution, and as expected the chosen threshold would classify only a minority of measures as anomalies. Even though the high computational cost of DeepAnT, the queue structure programmed with celery provided the correct control for the system, not compromising visualization and allowing further addition of others algorithms due to great scalability.

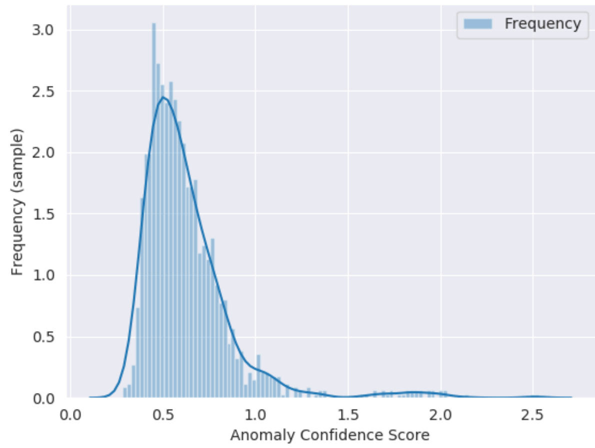


Figure 5. Distribution of anomaly score

The integration with Twilio API showed positive results for notification of anomalies and daily reports. The messages were programmed to include tips that could be used to avoid the problems that are happening, depending on the method and variable related to it. In Fig. 6 an example of a message can be seen.

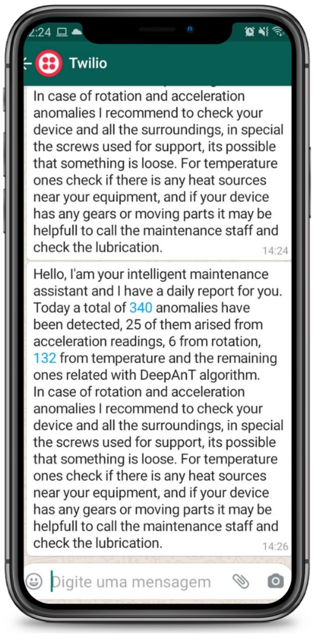


Figure 6. Example of a message sent by maintenance assistant

6. CONCLUSION

The popularization of PdM in industrial plants is of great importance, even though it faces its own challenges. This work presented an architecture for a PdM system using only unsupervised learning algorithms, so detection and analysis of sensed information do not need labeled data or be limited to specific types of machines present in training dataset. An illustrative example of the proposed architecture was implemented and validated. During the implementation the system used a smart sensor capable of multi-sensing, providing information about temperature and acceleration and angular speed in three-axis. Many technologies from the fourth industrial revolution were used, including IoT, cloud computing, artificial intelligence and smart devices. A deep learning method was applied together with two classical approaches for anomaly detection, and yet the proposed architecture provided scalability and flexibility for addition of new algorithms. The article showed the possibility of developing a PdM platform without supervised learning, however this class of algorithm has its own advantage. Performance evaluation is not presented in this work, since there are not benchmark dataset for the used sensor, therefore the application of a simulated system or a prototype for benchmark with anomalies created synthetically would be of great addition.

REFERENCES

Ahmad, S., Lavin, A., Purdy, S., and Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262, 134–147.

Amruthnath, N. and Gupta, T. (2018). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, 355–361. IEEE.

Cachada, A., Barbosa, J., Leitño, P., Gcraldcs, C.A., Deusdado, L., Costa, J., Teixeira, C., Teixeira, J., Moreira, A.H., Moreira, P.M., et al. (2018). Maintenance 4.0: Intelligent and predictive maintenance system architecture. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, 139–146. IEEE.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 1–58.

Cheng, F.T., Tieng, H., Yang, H.C., Hung, M.H., Lin, Y.C., Wei, C.F., and Shieh, Z.Y. (2016). Industry 4.1 for wheel machining automation. *IEEE Robotics and Automation Letters*, 1(1), 332–339.

Chiu, Y.C., Cheng, F.T., and Huang, H.C. (2017). Developing a factory-wide intelligent predictive maintenance system based on industry 4.0. *Journal of the Chinese Institute of Engineers*, 40(7), 562–571.

Espressif (2020). Esp 32. Available at `https://www.espressif.com/en/products/socs/esp32` (2020/01/05).

Giachino, J.M. (1986). Smart sensors. *Sensors and actuators*, 10(3-4), 239–248.

Hashemian, H.M. (2010). State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement*, 60(1), 226–236.

Huang, H.C., Lin, Y.C., Hung, M.H., Tu, C.C., and Cheng, F.T. (2015). Development of cloud-based automatic virtual metrology system for semiconductor industry. *Robotics and Computer-Integrated Manufacturing*, 34, 30–43.

Hunter, G.W., Stetter, J.R., Hesketh, P., and Liu, C.C. (2010). Smart sensor systems. *The Electrochemical Society Interface*, 19(4), 29.

INVENSENSE, I. (2013). Datasheet mpu6050, revision 3.4.

Lasi, H., Fettke, P., Kemper, H.G., Feld, T., and Hoffmann, M. (2014). Industry 4.0. *Business & information systems engineering*, 6(4), 239–242.

Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., and Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Computers in industry*, 57(6), 476–489.

Maier, A., Sharp, A., and Vagapov, Y. (2017). Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things. In *2017 Internet Technologies and Applications (ITA)*, 143–148. IEEE.

Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing.

Mobley, R.K. (2002). *An introduction to predictive maintenance*. Elsevier.

Munir, M., Siddiqui, S.A., Dengel, A., and Ahmed, S. (2018). Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7, 1991–2005.

Perera, C., Zaslavsky, A., Liu, C.H., Compton, M., Christen, P., and Georgakopoulos, D. (2013). Sensor search techniques for sensing as a service architecture for the internet of things. *IEEE Sensors Journal*, 14(2), 406–420.

Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., and Harnisch, M. (2015). Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, 9(1), 54–89.

Schwab, K. (2017). *The Fourth Industrial Revolution*. New York: Crown Business.

Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9), 1670–1679.

Taylor, S.J. and Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45.

Teng, M. (2010). Anomaly detection on time series. In *2010 IEEE International Conference on Progress in Informatics and Computing*, volume 1, 603–608. IEEE.

Twilio (2021). Twilio. Available at `https://www.twilio.com` (2020/01/05).

Velmurugan, R.S. and Dhingra, T. (2015). Maintenance strategy selection and its impact in maintenance function: A conceptual framework. *International Journal of Operations & Production Management*.

WSJ (2018). The impact of artificial intelligence on the world economy. Available at https://www.wsj.com/articles/the-impact-of-artificial-intelligence-on-the-world-economy-1542398991 (2020/01/05).