Opsi
Vol. 18, No. 1, June 2025
p-ISSN 1693-2102
e-ISSN 2686-2352

Utilizing machine learning for predictive maintenance of production machinery in small and medium enterprises

Yopa Eka Prawatya*, Noveicalistus H. Djanggu, Ratih Rahmahwati, Steven Lourensius
Industrial Engineering Department, Universitas Tanjungpura, Jalan Prof. Dr. H. Hadari Nawawi, Pontianak, Kalimantan Barat, Indonesia 78124

***Corresponding Author**: yopa.prawatya@industrial.untan.ac.id

Article history:
Received: 18 September 2024
Revised: 23 June 2025
Accepted: 26 June 2025
Published: 30 June 2025

Keywords:
Predictive maintenance
Machine learning
K-nearest neighbor
Decision tree
Random forest

ABSTRACT
Predictive maintenance involves the early detection of potential machine failures and subsequent maintenance to prevent such failures. Machine learning is a pertinent statistical method for predictive maintenance, enabling the early detection of machine failures and the implementation of preventive measures through a model. The development of the machine learning model commences with data collection from the machine, encompassing vibration, acceleration, machine temperature, and machine sound, facilitated by a microcontroller equipped with sensors. Subsequently, the data undergoes cleaning, including removing outliers or missing values and standardization. Data is partitioned into 70% allocated for training and 30% for testing. After determining hyperparameters and their values through hyperparameter tuning, the training data is utilized to train machine learning models, such as K-nearest neighbor, decision tree, and random forest models. Post-training, the models are evaluated using the remaining test data, employing performance metrics such as accuracy, precision, recall, and F1-score. The random forest model excels due to its utilization of a substantial number of trees for predictions and the full exploitation of the variables which F1-score is 91.22%. The best-performing model is subsequently deployed into a monitoring system, providing real-time machine condition predictions. The deployment results validate the accurate prediction of machine failures.

DOI:
https://doi.org/10.31315/opsi.v18i1.13479

1. INTRODUCTION

The advancement of the manufacturing sector is heavily reliant on the utilization of diverse machinery and equipment, aimed at bolstering efficiency, curbing costs, and expediting operational workflows. As such, the maintenance of these assets assumes critical significance. The challenges associated with maintenance management are not confined to large enterprises but also impinge on conventional businesses that may lack a comprehensive grasp of maintenance protocols. Beyond the dearth of maintenance comprehension, the reactive nature of maintenance exacerbates issues, with equipment often being serviced solely upon malfunction. This approach precipitates various ramifications including diminished output quality, processing velocity, and safety standards as equipment nears breakdown. Regular maintenance serves to sustain the optimal condition of machinery. Predictive maintenance stands out as an efficacious strategy to avert breakdowns. Grounded in prognostication tools leveraging historical data, statistical inference methodologies, and technical paradigms, predictive maintenance facilitates early identification of potential issues [1], [2]. In this regard, machine learning offers a powerful computational approach for processing large volumes of sensor and operational data commonly found in industrial settings. Through its ability to uncover complex patterns and adapt to new data, machine learning plays a critical role in solving various maintenance challenges, including fault prediction, anomaly detection, and maintenance schedule optimization [3], [4].

Predictive maintenance is considered more effective and efficient because maintenance is based on observations or monitoring and analysis of machine [5]–[7]. Predictive maintenance enables early detection of faults using predictive tools such as historical data (machine learning usage), integrity factors (such as visual aspects, wear and tear, and colour differences from the original), inferential statistical methods, and technical approaches [8], [9]. The advantages of predictive maintenance methods include the ability to know the condition of machines at any time, allowing operators to determine whether a machine needs to be repaired or not [10], [11]. The long-term use of predictive maintenance is highly economical [12], [13]. Through predictive maintenance, maintenance schedules can be controlled, spare parts inventory can be reduced, and machine damage can be minimized [14]–[16].

Machine condition monitoring is related to maintenance aspects based on the current and past condition of the machine [17], [18]. To determine the condition of a machine, sensors are installed around the machine to collect relevant information for analysis and decision-making regarding maintenance actions, so the machine can function well according to its original purpose [19]. During operation, machines provide a wealth of information in the form of sound, vibrations, temperature, lubricating oil condition, the quality and condition of the machine's electrical current, and the like [20]–[22].

Analysis of machine condition observations can be quickly obtained with the assistance of machine learning [23]. Machine learning is a computer algorithm development technology that utilizes fields of probability, statistics, artificial intelligence, information theory, psychology, control theory, philosophy, and computer science to mimic human intelligence [24]–[26]. Although machine learning provides a promising approach to predictive problems, the performance of machine learning heavily relies on the selection of appropriate machine learning methods [27]. The design process of a machine learning model for predictive maintenance involves several steps, including historical data selection, data preprocessing, model selection, model training and validation, and model maintenance [27], [28].

The research was conducted using supervised machine learning techniques, namely K-nearest neighbor, decision tree, and random forest. K-nearest neighbor (KNN) is a classification algorithm that uses labelled training data, which is divided into several classes, and this algorithm will classify new data that is fed into the existing groups [29], [30]. The study used an areca nut peeling machine as the research object, where the machine learning model was built based on its characteristics. The obtained machine learning model was applied to the operating machine, allowing for real-time predictions of the machine's condition. Through this research, a monitoring system was developed to oversee the machine's status and predict its condition. The research aimed to assist businesses, especially small and medium-sized enterprises, in implementing maintenance management in their production units to reduce costs incurred due to breakdowns. The variables measured on the equipment used included vibration acceleration, temperature, and sound.

2. MATERIALS AND METHODS

This research consists of several stages: data collection, machine learning model development, model performance verification, and implementation.

2.1. Data Collection

Condition monitoring data refers to measurements linked to the health status (indicators) of machines. These condition monitoring data are highly versatile and can include various types, such as vibration data, oil analysis data, acoustic data, temperature, pressure, moisture, humidity, weather data, or environmental data [32]. The data collected in this research are online data, including vibration, temperature, and sound. These three types of data are used because the characteristics of normal machine data and faulty machine data differ and can serve as good indicators. Data collection was carried out using sensors attached to the NodeMCU

ESP8266. The ADXL345 sensor was used to capture vibration data, the Type K Thermocouple sensor was used to capture temperature data, and the FC-04 sensor was used to capture sound data. The data were stored in CSV (comma-separated values) file format for further processing using the Python programming language.

(a) ADXL345 vibration sensor (b) Type K thermocouple sensor (c) FC-04 sound sensor

Figure 1. Sensor placement on the machine with the arrangement: (a) shows the ADXL345 vibration sensor placed on top of the engine body. The sensor is placed on top of the engine body so that it can record the overall vibration of the engine; (b) is the placement of the Type K Thermocouple sensor placed on top of the engine exhaust which is the heat exhaust from the engine; (c) shows the FC-04 sensor placed on top of the engine so that it can capture all the sounds produced by the engine.

2.2. Machine Learning Model Development

The collected data were then processed using Python programming language with Jupyter Notebook as the supporting software. Data processing began with the data preprocessing stage to prepare clean data for input into the machine learning model. Data exploration was also performed using various types of graphs to gain insight into the data characteristics. For the K-nearest neighbor method, the data were standardized first. The data processing was then continued by splitting the data into two parts: training data and test data [31]. Before feeding data to the machine learning model, the data must be free from missing values and outliers. The data is then trained, and hyperparameter tuning is performed to obtain high model accuracy. The selection of the best hyperparameters is done using the cross-validation method. In practice, the cross-validation method in machine learning has been widely used [33]. The most popular cross-validation method is called K-fold cross-validation, which can divide the training dataset into K folds. Suppose there are hyperparameters, then for each fold, it is studied using the remaining K1 folds as the training dataset and testing the model's performance on that fold. Then, continue by calculating the average performance over K folds. Hyperparameters are determined in the search process so that the average performance in cross-validation is maximized [33], [34]. After the data was prepared, the next step was the development of the machine learning model. The following is an input-process-output model in the development of a machine learning model for predictive maintenance in Figure 2.

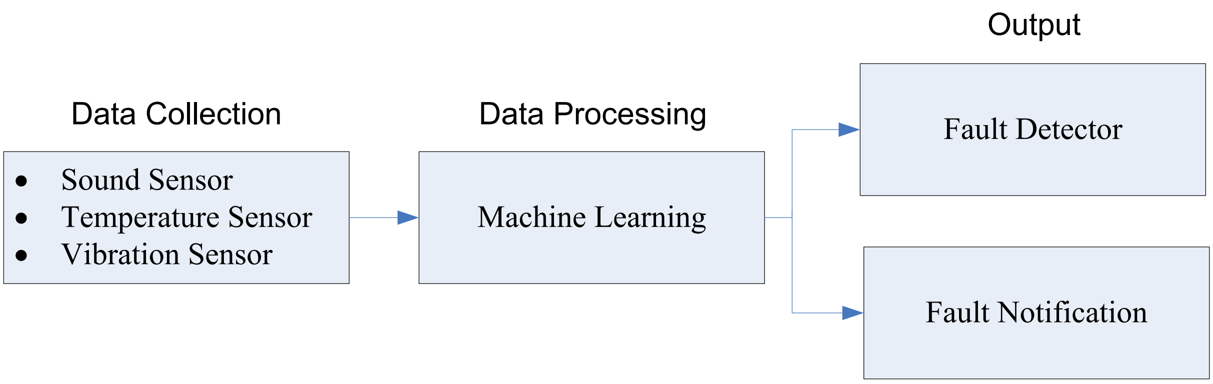


Figure 2. Diagram input-process-output

2.3. Machine Learning Performance Verification

Once the machine learning model was obtained, its performance was verified. The verification was carried out by testing whether the model produced the correct output and by measuring the performance of the machine learning model. Model performance was measured using three evaluation metrics: accuracy, precision, recall, and F1-score [36]. Precision is used to select the model that minimizes errors in classifying positive classes, while recall is used to select the model that minimizes errors in classifying negative classes. The F1-score is used to maximize the performance of both precision and recall. Data collection is repeated if the model does not produce satisfactory results [33], [34]. The best machine learning model is selected based on the highest average value obtained from the three-evaluation metrics.

2.4. Implementation (Model Deployment)

The best machine learning model obtained was implemented in real-time to detect the condition of equipment or machines. The implementation was carried out using NodeMCU ESP8266 to receive data from the sensors, which were then sent to a laptop for real-time processing by the designed program. The predicted machine condition output is displayed directly on the laptop processing the sensor data.

3. RESULTS

3.1. Data Preprocessing

The collected data includes vibration acceleration on the x-axis, vibration acceleration on the y-axis, vibration acceleration on the z-axis, machine temperature, and machine-generated sound. The machine data to be collected encompasses the characteristics of the machine when it is in normal condition and the characteristics of the machine when it is in an abnormal or damaged state. The NodeMCU ESP8266 was programmed to read data generated by sensors every 200 milliseconds or 0.2 seconds. The data successfully collected for normal machine conditions amounted to 2396 rows, and for failure conditions, there were 2300 rows of data.

The collected data will undergo a preprocessing stage. During this phase, the removal of missing values will take place. Missing values can occur due to errors in data reading by the microcontroller. Data collection errors can stem from two sources, namely sensor errors when capturing vibrations, temperature, or sound, or errors during the data recording process, particularly during encoding and decoding. Outliers in the data must be eliminated. Prior to eliminating outliers, the data will be standardized using the Z-Score. Data that is eliminated or categorized as outliers are those that exceed 1.5 standard deviations. Standardization of the data also serves to enhance the speed of training machine learning models. The training data is randomly selected, comprising 70% of the total available data, while the testing data is 30%. In addition to dividing the data into training and testing sets, the data is also split into two parts: predictor data and target data.

3.2. Exploratory Data Analysis

Exploratory data analysis is a data analysis process used in statistics to understand the basic characteristics of a dataset before conducting more in-depth analysis or statistical modelling. The main goal of EDA is to gain an initial understanding of the data, identifying patterns, relationships, anomalies, and other important properties that may be hidden in the dataset. Table 1 shows the average for each variable from normal machines and failure machines.

Table 1. Mean machine variables

Variable	Machine condition	
	Normal	Failure
Acceleration X	15.60622	16.06302
Acceleration Y	15.308746	16.109932
Acceleration Z	12.890949	13.230661
Temperature	106.512274	112.236182
Noise	122.137274	107.923077

Comparing characteristics between normal and failure machine data is shown by the average of variables from each class. In failure machines, the engine vibrates more strongly, and the resulting temperature is also higher, thus the average value of failure machines is higher compared to normal machines.

3.3. Model Training

Model training is the stage where data is fed into the model to be learned so that the model can make predictions based on the given data. On the other hand, hyperparameter tuning is the process of finding the combination of hyperparameters in the model that yields the highest accuracy. Table 2 shows the hyperparameter tuning results of each model.

Table 2. Model hyperparameters

Model	Hyperparameter	Value
KNN	n_neighbors	23
Decision Tree	max_depth	2
Random Forest	n_estimators	100
	max_features	2
	max_depth	9

The K-nearest neighbor model is a model that utilizes the Euclidean distance, and as such, data needs to be standardized to reduce the risk of errors and enhance computational speed. The K-nearest neighbor model has a hyperparameter called n_neighbors, which determines the number of nearest data points to be selected for making predictions. The range of n_neighbors used in hyperparameter tuning is from 1 to 150. The selection of the best hyperparameter is done using cross-validation.

In the K-nearest neighbor model, the optimal n_neighbors that provides the highest accuracy is found to be 23. This means that for each test data input, the model will choose 23 nearest data points to the test data, and the majority class among these 23 nearest data points will be assigned as the class for the test data. It's important to note that the hyperparameter n_neighbors must be an odd value to avoid ties in the nearest data points' classes, ensuring that one class doesn't have more instances than the others.

The decision tree model has a hyperparameter called max_depth, which represents the depth of a decision tree. The range of max_depth used for hyperparameter tuning is from 1 to 35. This means that 35 decision trees are generated, each with varying depths. The selection of the best tree is done using cross-validation. The hyperparameter max_depth that provides the best performance is a max_depth value of 2. This means that the decision tree model has a depth of 2 levels.

Figure 3 illustrates the model of the decision tree. This figure shows that the tree's root node is the variable "Machine_Temp" or machine temperature with a Gini index value of 0.495. The Gini index measures the purity of a variable. A smaller Gini index indicates a purer variable. A smaller Gini index makes it easier to distinguish between normal and faulty machine conditions. In the constructed decision tree model, the root node divides the data into two conditions: when the machine's temperature is greater than 109.125°C and the sound is less than 245, it is categorized as faulty. On the other hand, if the machine's temperature is less than 109.125°C, it is categorized as normal.

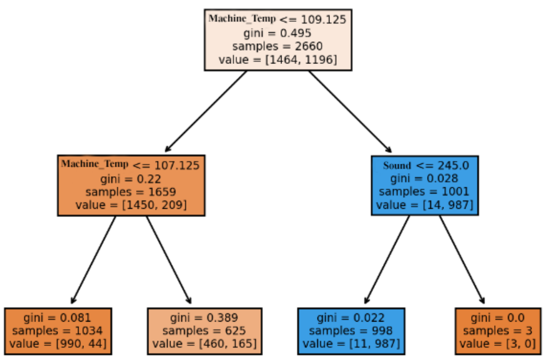


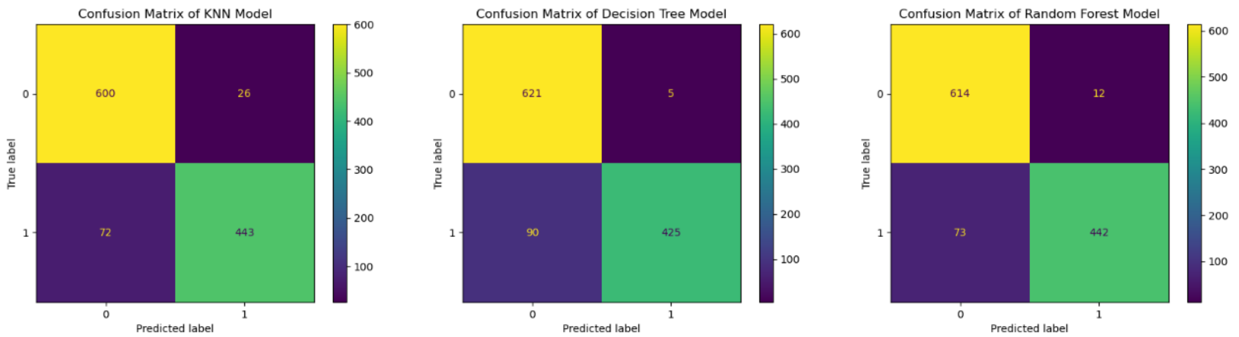
Figure 3. Decision tree model

3.4. Model Evaluation

Verification of the machine learning model is done using previously divided test data. Using the constructed model, predictions are made using the test data. The model's performance or prediction results can then be assessed through a confusion matrix. A confusion matrix is a matrix that maps correct and incorrect prediction results.

The evaluation of the performance of the three models is done using the F1-score metric. A good model is one that has an F1-score value close to 1 or equal to 1. The F1-score metric can effectively measure the performance of a machine learning model because it maximizes models with high precision and recall. This means that the model can accurately map the classes.

The prediction results of the K-nearest neighbor model, as shown in Figure 4(a), successfully predicted 600 normal machine conditions and 443 damaged conditions out of a total of 1,141 test data samples. This means that the model failed to predict 98 test data points, where 72 data points were predicted as normal machines but were actually damaged, and 26 data points were predicted as damaged machines but were actually normal. Calculations were performed for all models, and a recapitulation was carried out to select the model with the best performance.



(a) KNN model (b) Decision tree model (c) Random forest model

Figure 4. Confusion matrix for: (a) KNN model; (b) decision tree model; (c) Random forest model

The model selection is based on the highest F1-score value. Based on the calculation results, the random forest model provides the highest F1-score value, which is 91.22%. Therefore, the random forest model will be chosen for the implementation of the predictive maintenance system for fruit peeling machines.

Table 3. Model evaluation performance

Metrics	**Model**		
	KNN	**Decision Tree**	**Random Forest**
Accuracy	91.41%	91.67%	92.55%
Precision	94.45%	98.83%	97.35%
Recall	86.01%	82.52%	85.82%
F1-Score	90.03%	89.94%	91.22%

3.5. Model Deployment

The deployment process is the process of applying or implementing the best machine learning model into a system for prediction [36], [37]. The model chosen for implementation is the random forest model with an F1-score of 91.22%. The model is implemented into an integrated system consisting of data collection, processing, and prediction processes.

Figure 5 depicts the architecture of a monitoring system that integrates Internet of Things (IoT) components and machine learning to enable automated data-driven decision-making. The process is divided into three main stages :1) Microcontroller & Sensor: At this initial stage, various sensors (such as temperature, vibration, or sound sensors) are interfaced with a microcontroller to acquire real-time data from the physical environment. The microcontroller serves as a bridge that preprocesses and transmits the data to the next stage.

2) Data Storage: The data transmitted from the microcontroller is stored in a centralized or cloud-based database. This intermediate step ensures that data is systematically archived and readily available for further processing and analysis. 3). Storing & Prediction: In the final stage, the stored data is analyzed using machine learning algorithms deployed on servers or edge devices. This step enables pattern recognition, anomaly detection, or predictive analytics, ultimately supporting informed decision-making.

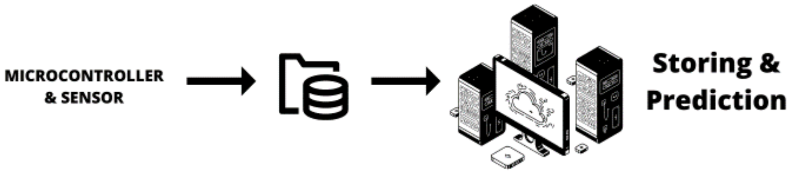


Figure 5. Monitoring system diagram [34]

To obtain data as input for the model to make predictions, a tool or medium is needed to collect data. In this research, a NodeMCU ESP8266 microcontroller along with sensors is used. Machine data is collected every 500 milliseconds and sent to a database. The data stored in the database must then be received or captured immediately to prevent it from being overwritten with new data. To receive this data, a program is created along with a dashboard for monitoring and supervision of the machines. The dashboard can monitor the machine's performance in real-time.

The dashboard will receive every sent data, which will be used for maintenance purposes. Each sent data will be fed directly into the machine learning model to obtain predictions regarding the machine's condition. On the dashboard, you can select the machine learning prediction display for data from temperature, vibration, and sound sensors. The dashboard for the fruit peeling machine used to predict the machine's condition is shown in Figure 6(a) and Figure 6(b) related to engine failure caused by temperature. Figure 4(a) displays the dashboard's interface for predicting normal machine conditions, while Figure 6(b) shows the dashboard's interface for predicting failure machine conditions.

(a) For normal condition (b) For failure condition

Figure 6. Display of the dashboard under: (a) normal condition; (b) failure condition

4. DISCUSSION

Predictive maintenance by predicting machine conditions based on machine characteristics can be done using machine learning models. To predict machine conditions using machine learning, data is first collected from a machine that can be used to predict machine conditions. In this research, variables such as machine vibrations, machine temperature, and machine sound are used. Data collection is done with the help of a microcontroller equipped with sensors. Data is collected from machines that are operating normally and machines that are in a damaged condition. The collected data is then cleaned of outliers and missing values to prepare it for input into a model. After data cleaning, the data characteristics are studied using a series of graphs. The data is then fed into a machine learning model. Hyperparameter tuning is also performed to obtain hyperparameters that yield the best model performance. The obtained models are then verified to determine their accuracy levels. The most crucial stage in predicting machine conditions using machine learning is deployment. The mentioned stages constitute the machine learning workflow, which consists of a series of steps starting from data collection, data preprocessing, model training and hyperparameter tuning, verification, and finally, model deployment.

The machine learning models used in this research are K-nearest neighbor, decision tree, and random forest, with F1-score values of 90.03%, 89.94%, and 91.22%, respectively. The random forest model is superior because it constructs a number of decision trees using all available data knowledge, and through the predictions of each decision tree, a voting process is performed, which becomes the output of the random forest model. Although the decision tree model provides high accuracy, it does not use all available variables for prediction, which may not fully maximize the potential of the data for prediction. Additionally, decision tree predictions can be localized optima. The model with the best accuracy is then selected for application (deployment) into a system. The model is applied to a program that receives input variables such as machine vibration acceleration, machine temperature, and machine sound collected using a microcontroller. These variables serve as input for the model to make predictions. The designed program not only predicts machine conditions but also serves as a monitoring system for the machines. The application results show that the model can effectively predict machine damage as well as normal machine conditions.

5. CONCLUSION

In conclusion, this study successfully demonstrates the application of machine learning models to predictive maintenance by predicting machine conditions based on vibration, temperature, and sound data collected from machines in both normal and faulty states. Data preprocessing, including the removal of missing values and outliers, along with standardization, was crucial for preparing the dataset for model training. The study employed K-nearest neighbor, decision tree, and random forest models, with the random forest model achieving the highest F1-score of 91.22%, making it the optimal choice for deployment in a predictive maintenance system.

The deployment process integrated the random forest model into a real-time monitoring system, using a NodeMCU ESP8266 microcontroller to collect data every 500 milliseconds. The model's implementation in a dashboard system enables real-time predictions of machine conditions, enhancing the effectiveness of maintenance strategies. The results indicate that machine learning models, particularly the random forest, are effective in predicting machine conditions, offering significant potential for improving maintenance practices and minimizing machine downtime.

Further research is expected to explain the division of categories of engine damage levels (low, medium or severe) based on the time interval between engine damages so that the time between damages will be useful in predicting the remaining service life of the engine or the time when engine maintenance must be carried out.

REFERENCES

[1] R. Quatrini, E., Costantino, F., Di Gravio, G., Patriarca, "Condition-based maintenance-an extensive literature review," *Machines*, vol. 8, no. 31, pp. 1–28, 2020.

[2] G. Zonta, T., da Costa, C., da Rosa,R., Jose de Lima, M., da Trindade,E., Pyng Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Computers & Industrial Engineering*, vol. 150, pp. 1–17, 2020.

[3] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, 2015, doi: 10.1109/TII.2014.2349359.

[4] L. Puryani., Chalida,N., Soepardi,A., Chaeron,M., Nafisah, "Improving the job shop scheduling algorithm to minimize total penalty cost considering maintenance activity," *OPSI*, vol. 17, no. 2, pp. 355–367, 2024, doi: https://doi.org/10.31315/opsi.v17i2.12291.

[5] M. M. Ardhianto and R. Sumarwanto, "Analisa predictive berbasis supervised machine learning terhadap kerusakan peralatan pembangkit," *Jurnal Teknologi Elekterika*, vol. 19, no. 2, pp. 143–148, 2022.

[6] V. Bahrudin,A., Setiawan,I., Arifin,M.M., Fipiana,W.I., Lusia, "Analysis of preventive maintenance breakdown maintenance on production achievement in the food seasoning industry," *OPSI*, vol. 14, no. 2, pp. 253–261, 2021, doi: https://doi.org/10.31315/opsi.v14i2.5540.

[7] J. Mahendra,I.G.O., Atmaji,F.T.D., Alhilman, "Proposes maintenance policy design on paving molding machine using Reliability and Risk Centered Maintenance (RRCM) method at PT XYZ," *OPSI*, vol. 14, no. 2, pp. 231–243, 2021, doi: https://doi.org/10.31315/opsi.v14i2.5477.

[8] 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 and Industrial Engineering*, vol. 137, no. August, 2019, doi: 10.1016/j.cie.2019.106024.

[9] M. & M. M. Pandey, J.S., Ahn, C. Lee, A., "Recent advances in the application of natural fiber based composites," *Macromolecular Materials and Engineering*, vol. 295, no. 11, pp. 975–989, 2019.

[10] M. Montero-Jimenez, J.J., Scwartz, S., Vingerhoeds, R., Grabot, B., Salaun, "Towards multi-model approaches to predictive maintenance: a systematic literature survey on diagnostics and prognostics," *Journal Manufacturing System*, vol. 56, pp. 536–557, 2020.

[11] S. Dayo-Olupona,O., Genc, B., Celik, T., Bada, "Adoptable approaches to predictive maintenance in mining industry: An overview," *Resources Policy*, vol. 86, p. 104291, 2023.

[12] A. P. Sahba, R., Radfar, R., Ghatari, A.R., Ebrahimi, "Development of industry 4.0 predictive maintenance architecture for broadcasting chain," *Advanced Engineering Informatics*, vol. 49, pp. 1–11, 2021.

[13] D. Pratiwi,A.I., Sihombing,B., Sayuti,M., Pradana,N., Adetia, "DMB machine reliability analysis through failure mode effect analysis and reliability block," *OPSI*, vol. 14, no. 1, pp. 81–88, 2021, doi: https://doi.org/10.31315/opsi.v14i1.4745.

[14] H. B. Shin, J.H., Jun, "On condition based maintenance policy," *Journal Computational Design and Engineering*, vol. 2, pp. 119–127, 2015.

[15] S. S. Zhang, Y., Ren, S., Liu Y., "A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products," *Journal Clean Production*, vol. 142, pp. 626–641, 2017.

[16] NH. Wang, Y., Gogu C., Binaud, N., Bes C., Haftka RT., Kim, "A cost-driven predictive maintenance policy for structural airframe maintenance," *Chinese Journal of Aeronautics*, vol. 30, no. 3, pp. 1242–1257, 2017.

[17] A. R. Mohanty, *Machinery Condition Monitoring: Principles and Practices*. Boca Raton: CRC Press, 2014.

[18] B. Putra,S.S.W., Rahmawati, "Raw mill machine effective measurement through the Total Production Maintenance (TPM) Implementation," *OPSI*, vol. 15, no. 2, pp. 153–163, 2022, doi: https://doi.org/10.31315/opsi.v15i2.7729.

[19] Felecia, "Fuzzy logic reliability centered maintenance," *Jurnal Teknik Industri*, vol. 16, no. 2, pp. 121–126, 2014.

[20] K. B. Jayaswal, P., Wadhawani, A.K., Mulchandani, "Machine fault signature analysis," *International Journal of Rotating Machinery*, vol. 2008, pp. 1–10, 2008.

[21] E. Zhu, J., He, D., Bechhoefer, "Survey of lubrication oil condition monitoring, diagnostics and prognostics techniques and systems," *Journal of Chemical Science and Technology*, vol. 2, no. 3, pp. 100–115, 2013.

[22] R. Gangsar,P., Tiwari, "Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art-review," *Mechanical Systems and Signal Processing*, vol. 144, pp. 1–37, 2020.

[23] A. Schwendemann, S., Amjad, Z., Sikora, "A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grindings machines," *Computers in Industry*, vol. 125, pp. 1–17, 2021.

[24] I. El Naqa and M. J. Murphy, "Machine learning in radiation oncology," *Machine Learning in Radiation Oncology*, pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3.

[25] J. Q. K.Wang., J.Xu., S. Zhang., Tan, "Towards low-budget energy efficiency design in additive manufacturing based on variational scale-aware transfromer," *Clean Production*, vol. 393, pp. 136–168, 2023.

[26] T.M.Mitchell, *Machine learning*. New York: McGraw-Hill, 2007.

[27] 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 and Industrial Engineering*, vol. 137, no. August, 2019, doi: 10.1016/j.cie.2019.106024.

[28] I. Arena, S., Florian, E., Sgarbossa, F., Sølvsberg, E., Zennaro, "A conceptual framework for machine learning algorithm selection for predictive maintenance," *Engineering Applications of Artificial Intelligence*, vol. 133, pp. 1–15, 2024.

[29] Y. Chen, C., Fu, H., Zheng, Y., Tao, F., Liu, "The advance of digital twin for predictive maintenance: The role and function of machine learning," *Journal of Manufacturing Systems*, vol. 71, pp. 581–594, 2023.

[30] S. Ray, "A quick review of machine learning algorithms- A review," in *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, 2019, pp. 35–39.

[31] F. Chac´on, A.M., Ramírez, I.S., M´arquez, "K-nearest neighbour and K-fold cross-validation used in wind turbines for false alarm detection," *Sustainable Futures*, vol. 6, pp. 100–132, 2023.

[32] D. Das, Oguzhan., Das-Bagci, Duygu., Birant, "Machine learning for fault analysis in rotating machinery: A comprehensive review," *Heliyon*, vol. 9, pp. 1–34, 2023.

[33] N. Z. Wang,B., Gong, "Stealing hyperparameters in machine learning," in *IEEE Symposium on Security and Privacy*, 2018, pp. 36–52.

[34] C. Ashmore,R., Calinescu, R., Paterson, "Assuring the machine learning lifecycle: desiderata, methods and challenges," *ACM Computing Surveys*, vol. 54, no. 5, pp. 111–139, 2021.

[35] Y. Peksen, M.F., Yurtsever,U., Uyaroglu, "Enhancing electrical panel anomaly detection for predictive maintenance with machine learning and IoT," *Alexandria Engineering Journal*, vol. 96, pp. 112–123, 2024.

[36] J. Díaz-Ramírez, "Machine learning and deep learning," *Ingeniare*, vol. 29, no. 2, pp. 182–183, 2021.

[37] B.Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, 2018.

[38] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons*, vol. 4, pp. 51–62, 2017.