Machine Learning Applied to Industrial Machines for an Efficient Maintenance Strategy: A Predictive Maintenance Approach

Bruno Mota, Pedro Faria(✉), and Carlos Ramos

GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, LASI; Polytechnic of Porto, Rua Dr. António Bernardino de Almeida 431, 4200-072 Porto, Portugal
{bamoa,pnf,csr}@isep.ipp.pt

Abstract. Maintenance activities are crucial in manufacturing environments to reduce machine breakdowns and maintain product quality. However, traditional maintenance strategies can be expensive, as they can lead to unnecessary maintenance activities. As a result, Predictive Maintenance (PdM) can be a great way to solve these issues, as it enables the prediction of a machine's condition/lifespan allowing for maintenance-effective manufacturing. This paper aims to address these issues by proposing a novel methodology to improve the performance of PdM systems, by proposing a machine learning training methodology, an automatic hyperparameter optimizer, and a retraining strategy for real-time application. To validate the proposed methodology a random forest and an artificial neural network model are implemented as well as explored. A synthetic dataset, that replicates industrial machine data, was used to show the robustness of the proposed methodology. Obtained results are promising as the implemented models can accomplish up to 0.97 recall and 93.15% accuracy.

Keywords: Data Preprocessing · Hyperparameter Optimization · Predictive Maintenance

1 Introduction

Several changes have been happening in the energy sector, namely with the implementation of the smart grid concept [1], having more active participation of electricity consumers in demand response programs [2, 3]. Booming innovation in Big data, data analytics, and Internet of Things (IoT) has resulted in a shift in traditional industrial maintenance strategies to systems capable of forecasting machine lifespan [4, 5]. Furthermore, taking into account energy usage is also critical for optimizing production lines in industrial environments, because machine health can have a significant impact on a machine's energy efficiency capabilities [6, 7]. Accordingly, it is in these industries' best interests to implement these systems to minimize energy consumption, reducing not only costs but also contributing to a sustainable future through energy savings. There has been the development of two new maintenance concepts for detecting abnormalities in the production environment: prognostic and health management, as well as condition-based maintenance [8]. Predictive Maintenance (PdM), which analyzes past data to forecast behavior patterns, is frequently used with these two principles in mind, either with prognosis and health management or condition-based maintenance, and in some circumstances, the application of both [9]. The use of predictive systems to determine when maintenance activities are required is critical in a manufacturing environment, not only to avoid wasteful expenses and cut potential Greenhouse gas emissions but also to enhance product quality. According to [10], maintenance expenses can represent 15% to 70% of the cost of manufactured products. Predictive maintenance enables continuous monitoring of the machine's integrity, allowing maintenance to be performed only when absolutely needed, reducing unnecessary maintenance costs. Moreover, PdM prevents, to some extent, machine breakdowns, which are responsible for the emission of Greenhouse gas emissions in some industrial sectors [11]. Prediction systems that use statistical inference, historical data, engineering methods, and integrity factors allow for early abnormalities detection [12]. Forecasting a machine's condition and/or lifespan can be done through a variety of techniques, such as Artificial Neural Networks (ANNs) [13–15], Random Forests (RFs) [16–18], deep learning [19], digital twins [20], support vector machines [21], k-means [22], gradient boosting [23], naive bayes [24], and decision trees [25]. Other noteworthy techniques are presented in [12] as well as in [26].

This paper focuses on the implementation as well as the exploration of the advantages and disadvantages of the two most popular machine learning approaches, according to [12], for PdM: ANNs (27% model employment) and RFs (33% model employment). The prominent use of RF in PdM systems, due to its performance and easy implementation, is the main reason for the exploration of this model in the present paper. Nevertheless, an ANN model has the potential to outperform an RF, both in recall and context adaption, when its hyperparameters are adequately optimized. Furthermore, unlike the RF model, ANNs have the advantage of backpropagation (i.e., fine-tuning of the network's weights based on the error rate), allowing a current model to be constantly fed with data and improve over time without the need to recreate the model every time there is new training data, which is ideal for manufacturing environments.

The work in [13] proposes an ANN for PdM using the mean time to failure values and backpropagation for adjusting the neuron's weights. A PdM system for air booster compressor motors is proposed in [14] that employs an ANN with optimized weights and bias by using a particle swarm optimization algorithm. Also using an ANN, the proposed work in [15] focuses on a PdM system for induction motors that optimizes hyperparameters (e.g., number of hidden layers and neurons) to improve performance in the model. For RF, the work in [16] proposes a real-time PdM system for production lines using IoT data. A new PdM methodology, using RF, is proposed in [17] to allow dynamic decision rules to be imposed for maintenance management. A data-driven PdM system applied to woodworking is proposed in [18] using an RF that takes advantage of event-based triggers.

Of the above-cited works, none tackle, to the extent of the present paper, the main problem plaguing PdM problems, imbalanced data. Furthermore, with the exception of the works in [14] and [15], there is little to no optimization regarding hyperparameters, which can improve model performance significantly primarily in imbalanced datasets.

Finally, only the work in [16] considers real-time deployment and only the work in [13] takes advantage of the backpropagation feature for retraining. As such, the premise of this paper is to contribute to the progression of the current state-of-art by proposing:

- An innovative machine learning training approach for PdM that aims to improve model performance while also taking into account imbalanced and irrelevant/erroneous data.
- An automatic hyperparameter optimization strategy, used to determine the optimal hyperparameters for the ANN and RF, hence enhancing the models' performance even further.
- The application in real-time of both implemented models, by taking into account model retraining and user application.

This paper structure is divided into five sections. After this introductory and state-of-art section, Sect. 2 describes the training and testing dataset used to validate the proposed methodology. Section 3 describes the proposed methodology for PdM on an ANN and RF, while Sect. 4 presents the obtained results of the implemented models, as well as a discussion regarding such a topic. The conclusions are presented in Sect. 5.

2 Training/Testing Dataset

The PdM dataset used for training and testing of the proposed methodology was made available from the University of California in Irvine, Machine Learning Repository [27].

The PdM dataset from 2020, labeled "AI4I 2020 Predictive Maintenance Dataset Data Set," is freely accessible in [28]. The synthetic dataset has 10,000 data points where 339 represent failures and 9661 non-failures data points (i.e., a ratio of 1:28), as presented in Fig. 1. The machine data is the following:

- Air temperature–defines the exterior temperature of the machine, in Kelvin (K);
- Process temperature – defines the temperature produced within the machine, in Kelvin (K);
- Rotational speed–defines the rotational speed of the tools inside the machine, in Revolutions per minute (rpm);
- Torque–defines the force required to rotate the machine's tools, in Newton-meters (Nm);
- Tool wear–defines the amount of deterioration of the tools inside the machine, in minutes until breakdown (min);
- Machine failure–defines a machine failure status by assuming the value 0 for non-failure and 1 for failure.

The correlation heatmap between the used dataset features is described in Fig. 2.

It demonstrates that there is a medium positive correlation between machine failure and the features torque (0.190 positive correlation) and tool wear (0.110 positive correlation). On the other hand, the lowest correlation found to machine failure was the rotational speed (0.044 negative correlation) and process temperature (0.036 positive correlation).

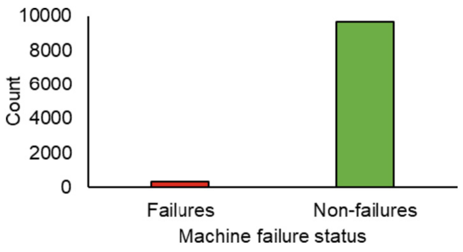
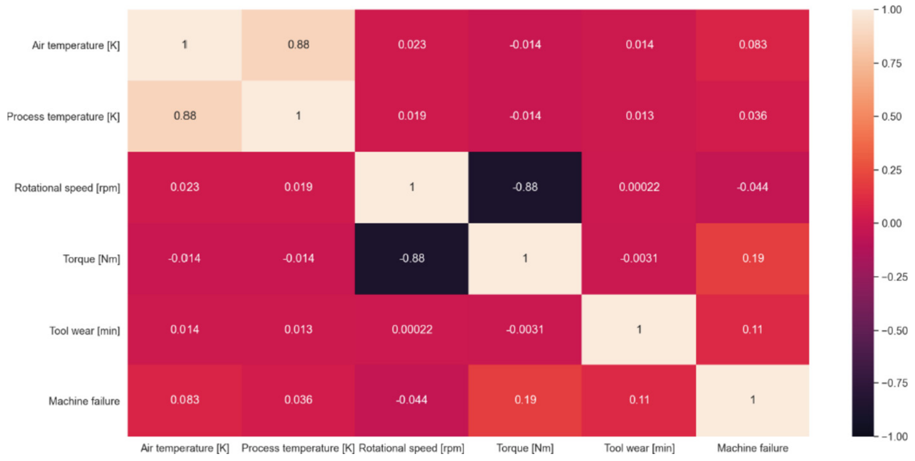


Fig. 1. Machine failure status bar chart of the used dataset.



	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure
Air temperature [K]	1	0.88	0.023	-0.014	0.014	0.083
Process temperature [K]	0.88	1	0.019	-0.014	0.013	0.036
Rotational speed [rpm]	0.023	0.019	1	-0.88	0.00022	-0.044
Torque [Nm]	-0.014	-0.014	-0.88	1	-0.0031	0.19
Tool wear [min]	0.014	0.013	0.00022	-0.0031	1	0.11
Machine failure	0.083	0.036	-0.044	0.19	0.11	1

Fig. 2. Correlation heatmap between the used dataset features.

3 Proposed Methodology

Two machine learning models, an ANN and an RF model, are implemented and explored for PdM. In the proposed methodology, training of the implemented models can be done in batches, mini-batches, or continuous data streaming. Before real-time training, an initial model is constructed through a dataset, and only then, the training process is carried out in real-time via data streaming or mini-batches.

The initial model for the ANN or RF is constructed using:

- The dataset described in Sect. 2;
- The Holdout method, 80% for training and 20% for testing;
- A Min-Max approach for data normalization;
- A newly added dataset feature, machine temperature difference (i.e., process temperature − air temperature), replaces the process and air temperature features. It focuses on improving model performance, by reducing the number of inputs for less complexity and better correlation between temperature and machine failure;
- A data balancing method, 5% oversampling on failure data and a majority undersampling strategy on non-failure data. To achieve this, the imbalanced-learn [29] library was used;

- A 5-fold cross-validation splitting strategy to search for the best hyperparameters.

It is worth mentioning that, the Holdout method and the 5-fold cross-validation splitting strategy described above were employed as safe measures to prevent overfitting of the models.

The proposed methodology for real-time training begins by obtaining the most recent machine data, described in Sect. 2, from machine databases in the facility. Afterward, prior to training, a data preprocessing phase is employed, which can be divided into six sequential subphases:

1. Data aggregator–combines all acquired data into a single data file;
2. Data normalization–standardizes data units and types between machines, using a Min-Max technique with the MinMaxScaler method [30] from the Scikit-Learn library [31];
3. Data imputation–fills missing values on the obtained data, through a k-Nearest Neighbors imputation approach with the KNNImputer method [32] from the Scikit-Learn library;
4. Data filtering–removes any potentially incorrect or irrelevant data, by detecting outliers using the Z-score with the SciPy stats Z-score method [33] from the SciPy library [34];
5. Data engineering–creates or removes features to better depict the underlying problem;
6. Data balancing–balances machine data failure and non-failure points, with the imbalanced-learn library [29].

Then, the preprocessed data is used to train the machine learning models (i.e., ANN or RF), wherein the ANN neuron weights are adjusted due to the back-propagation feature, or, in the case of the RF, the model has to be reconstructed from the start using the new and past data.

The methodology for real-time application of the implemented machine learning models in a machine can be divided into three phases:

1. Data acquisition–obtains the necessary machine data from the machine to be inspected;
2. Data preprocessing–applies data normalization, imputation, filtering, and engineering on the obtained data;
3. Machine failure status prediction–uses one of the models, designated by the user, to predict the machine failure status (0 for non-failure and 1 for failure).

3.1 Artificial Neural Network Training

The ANN was trained using an automatic hyperparameter optimizer, which focuses on finding the optimal hyperparameter values to obtain a high-performing model. This is achieved by using the GridSearchCV [35] method available from the Scikit-Learn library. The automatic hyperparameter optimizer works by exploring each hyperparameter's possible values, at random, in order to find a high-performing ANN model, which contains the optimal values for each hyperparameter. Table 1 presents the possible and found optimal hyperparameter values for the ANN model. However, some hyperparameters were predefined, as there was no need to find the optimal value, such as the loss function, defined with binary cross-entropy function, metrics with binary accuracy, the number of input neurons as 4 (temperature difference, rotational speed, torque, tool wear), the number of output neurons as 1 (machine failure), the output layer activation function defined with a sigmoid function, and a normal weight initialization in the hidden layers.

For the ANN classifier, the KerasClassifier [36] from the Keras [37] library was used. It operates through rules created during the training phase to achieve the lowest possible accuracy error in contrast to the training classes. The model is ready to generate predictions once it has been properly fitted using training data.

Table 1. Artificial neural network hyperparameters possible and optimal values.

Hyperparameter	Possible Values	Optimal Value
Batch Size	10, 20, 40, 60, 80, 100, 200, 500, 1000, 2000, or 5000	5000
Dropout Regularization on Hidden Layers	0%, 5%, 10%, 20%, 30%, 35%, 40%, 50%, 60%, 70%, 80%, or 90%	35%
Dropout Regularization on Input Layer	0%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, or 90%	0%
Epochs	10, 50, 100, 150, 200, 300, 500, 1000, 2000, 5000, 8000, or 10000	5000
Neuron Activation Function	Hard Sigmoid, Linear, Relu, Sigmoid, Softmax, Softplus, Softsign, or Tanh	Relu
Hidden Layers Neuron Composition	1 to 4 layers 5, 10, 15, 20, 25, or 30 neurons per layer	25 neurons in layer 1 20 neurons in layer 2 15 neurons in layer 3 15 neurons in layer 4
Optimizer	Adadelta, Adagrad, Adam, Adamax, Nadam, RMSprop, or SGD	Nadam
Weight Initialization in Input Layer	Glorot Normal, Glorot Uniform, He Normal, He Uniform, Lecum Uniform, Normal, Uniform, or Zero	Glorot Uniform

3.2 Random Forest Training

The RF was also trained using an automatic hyperparameter optimizer, aiming to find a robust RF model. This is accomplished through the RandomizedSearchCV [38] method. This method focuses on determining the optimal estimator to employ in the model by selecting one of the possible values for each hyperparameter at random and then assessing each estimator based on their accuracy scores. Each hyperparameter's possible values and optimal value for the RF model are shown in Table 2. The RandomForestClassifier [39] was used as the RF classifier.

Table 2. Random forest hyperparameters possible and optimal values.

Hyperparameter	Possible Values	Optimal Value
Bootstrap Sample	True or False	True
Criterion Function	Gini or Entropy	Gini
Max Depth	10 to 32	10
Max Features	Auto, Sqrt, or Log2	Log2
Min Samples Leaf	1, 2, 4, 6, 8, or 10	1
Min Samples Split	2, 5, 10, 20, or 30	2
Tree Amount	200 to 3000	511

4 Results and Discussion

Four metrics were used to validate the performance of the proposed machine learning models: recall, precision, f1-score, and accuracy. It is worth noting that, since PdM problems commonly have very imbalanced datasets that have a low number of failure data points, the recall metric was considered to be the most relevant performance metric to validate the proposed methodology. The ANN performance metrics using the optimal hyperparameters found in Table 1 and the performance of the RF model using the optimal hyperparameters in Table 2 are shown in Table 3.

Table 3. Performance metrics of the proposed machine learning models using their respective optimal hyperparameters.

Machine Learning Model	Recall	Precision	F1-score	Accuracy
Artificial Neural Network	0.97	0.15	0.27	83.65%
Random Forest	0.95	0.30	0.46	93.15%

According to the results presented in Table 3, each model has its own benefits and drawbacks, with the ANN being slightly better at predicting when there is about to be a machine breakdown, since it has the highest recall, and the RF excelling at lowering the number of false alarms (i.e., false positives), because of having the highest precision and accuracy scores. As a result, on one hand, if maintenance costs are inexpensive and undetected machine breakdowns can lead to dire consequences, the ANN is the preferred model to be employed. On the other hand, the RF model is better at reducing the number of false alarms, which reduces unnecessary maintenance activities when compared to the ANN. Nevertheless, both models have good accuracy scores, mainly the Random Forest model with 93.15%, for this type of problem, where imbalanced predictive maintenance datasets are common and negatively affect accuracy scores. Table 4 presents the ANN and RF confusion matrixes. It is noteworthy that there is a big trade-off between true positives and false positives between the two models, with the RF only having 1 more unsuccessful machine failure prediction but having 191 fewer false alarms than the ANN. Therefore, in general, even though the recall was considered to be the most relevant metric, the RF model has the best performance overall, since it does not fall behind too much on recall and all other metrics are much better than in the ANN model. It is worth mentioning that another work [40] utilized the same dataset as the current paper to justify the usage of a bagged trees ensemble classifier. However, cited work did not split the dataset for training and testing, resulting in an overfitted model and inflated results, because of this, no comparison was made to this work. Despite the fact that the cited work inflated their obtained results, it achieved a recall score of only 0.71, lower than the present paper's ANN model with a recall of 0.97 and RF with 0.95.

Table 4. Artificial neural network and random forest confusion matrix.

Predicted		Actual	
		Failure	Non-failure
Artificial Neural Network	Failure	59	325
	Non-failure	2	1614
Random Forest	Failure	58	134
	Non-failure	3	1805

5 Conclusion

To further reduce costs and improve product quality, the manufacturing industry has been investing in PdM strategies to cut down on unnecessary maintenance costs, as PdM systems are capable of predicting machine condition/lifespan allowing for maintenance-effective manufacturing.

The proposed methodology aims to improve performance in machine learning models for PdM problems by proposing a novel training methodology, an automatic hyper-parameter optimization strategy, and a new retraining method. To achieve this, an ANN and RF models are implemented and explored. A synthetic dataset for PdM, containing imbalanced data, is presented to validate the proposed methodology.

The obtained results show the robustness of the proposed methodology, with the ANN model accomplishing a recall of 0.97, a precision of 0.15, an f1-score of 0.27, and an accuracy of 83.65%. The RF model was able to excel even further by achieving a lower recall of 0.95, but having a much better precision of 0.30, an f1-score of 0.46, and an accuracy of 93.15%. In general, the RF model has better performance overall, nevertheless, it is clear that the ANN is slightly better at reducing true positives while the RF reduces false positives.

Future work will address the use of real-world data instead of a synthetic dataset, allowing to better evaluate the effectiveness of the proposed methodology in practical manufacturing environments. In addition model interpretability, through eXplainable Artificial Intelligence (XAI), will also be explored for the proposed ANN and RF models, in order to improve confidence in PdM systems.

Funding. This work has received funding from project MUWO (grant agreement No. 771066), through COMPETE program, under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF/FEDER). Pedro Faria has Support from FCT trough project CEECIND/01423/2021. The authors acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020) to the project team.

References

1. Ramos, D., Faria, P., Gomes, L., Vale, Z.: A contextual reinforcement learning approach for electricity consumption forecasting in buildings. IEEE Access. **10**, 61366–61374 (2022). https://doi.org/10.1109/ACCESS.2022.3180754
2. Faria, P., Vale, Z.: Distributed energy resource scheduling with focus on demand response complex contracts. J. Mod. Power Syst. Clean Energy **9**, 1172–1182 (2021). https://doi.org/10.35833/MPCE.2020.000317
3. Mashal, I., Khashan, O.A., Hijjawi, M., Alshinwan, M.: The determinants of reliable smart grid from experts' perspective. Energy Informatics. **6**, 1–23 (2023). https://doi.org/10.1186/S42162-023-00266-3/TABLES/5
4. Sharma, A., Yadava, G.S., Deshmukh, S.G.: A literature review and future perspectives on maintenance optimization (2011). https://doi.org/10.1108/13552511111116222
5. Faccio, M., Persona, A., Sgarbossa, F., Zanin, G.: Industrial maintenance policy development: a quantitative framework. Int. J. Prod. Econ. **147**, 85–93 (2014). https://doi.org/10.1016/j.ijpe.2012.08.018
6. Mota, B., Gomes, L., Faria, P., Ramos, C., Vale, Z., Correia, R.: Production line optimization to minimize energy cost and participate in demand response events. Energies (Basel). **14**, 462 (2021). https://doi.org/10.3390/en14020462
7. Ramos, C., Barreto, R., Mota, B., Gomes, L., Faria, P., Vale, Z.: Scheduling of a textile production line integrating PV generation using a genetic algorithm. Energy Rep. **6**, 148–154 (2020). https://doi.org/10.1016/j.egyr.2020.11.093
8. Garg, A., Deshmukh, S.G.: Maintenance management: literature review and directions (2006). https://doi.org/10.1108/13552510610685075
9. Shin, J.H., Jun, H.B.: On condition based maintenance policy. J. Comput. Des. Eng. **2**, 119–127 (2015). https://doi.org/10.1016/j.jcde.2014.12.006
10. Thomas, D.S.: The Costs and Benefits of Advanced Maintenance in Manufacturing, pp. 1–45. National Institute of Standards and Technology (2018). https://doi.org/10.6028/nist.ams.100-18
11. Rodriguez, P.C., Marti-Puig, P., Caiafa, C.F., Serra-Serra, M., Cusidó, J., Solé-Casals, J.: Exploratory analysis of SCADA data from wind turbines using the K-means clustering algorithm for predictive maintenance purposes. Machines **11**, 270 (2023). https://doi.org/10.3390/machines11020270

12. Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., Francisco, R.daP., Basto, J.P., Alcalá, S.G.S.: A systematic literature review of machine learning methods applied to predictive maintenance. Comput. Ind. Eng. **137**, 106024 (2019). https://doi.org/10.1016/j.cie.2019.106024
13. Koca, O., Kaymakci, O.T., Mercimek, M.: Advanced predictive maintenance with machine learning failure estimation in industrial packaging robots. In: Proceedings of the 2020 15th International Conference on Development and Application Systems (DAS 2020), pp. 1–6. Institute of Electrical and Electronics Engineers Inc. (2020). https://doi.org/10.1109/DAS49615.2020.9108913
14. Rosli, N.S., Ain Burhani, N.R., Ibrahim, R.: Predictive maintenance of air booster compressor (ABC) motor failure using artificial neural network trained by particle swarm optimization. In: 2019 IEEE Student Conference on Research and Development (SCOReD 2019), pp. 11–16. Institute of Electrical and Electronics Engineers Inc. (2019). https://doi.org/10.1109/SCORED.2019.8896330
15. Kavana, V., Neethi, M.: Fault analysis and predictive maintenance of induction motor using machine learning. In: 3rd International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT 2018), pp. 963–966. Institute of Electrical and Electronics Engineers Inc. (2018). https://doi.org/10.1109/ICEECCOT43722.2018.9001543
16. Ayvaz, S., Alpay, K.: Predictive maintenance system for production lines in manufacturing: a machine learning approach using IoT data in real-time. Expert Syst. Appl. **173**, 114598 (2021). https://doi.org/10.1016/j.eswa.2021.114598
17. Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., Loncarski, J.: Machine learning approach for predictive maintenance in Industry 4.0. In: 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA 2018). Institute of Electrical and Electronics Engineers Inc. (2018). https://doi.org/10.1109/MESA.2018.8449150
18. Calabrese, M., et al.: SOPHIA: an event-based IoT and machine learning architecture for predictive maintenance in Industry 4.0. Information (Switzerland) **11**, 202 (2020). https://doi.org/10.3390/INFO11040202
19. Nguyen, K.T.P., Medjaher, K.: A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliab. Eng. Syst. Saf. **188**, 251–262 (2019). https://doi.org/10.1016/j.ress.2019.03.018
20. Haghshenas, A., Hasan, A., Osen, O., Mikalsen, E.T.: Predictive digital twin for offshore wind farms. Energy Informatics **6**, 1–26 (2023). https://doi.org/10.1186/S42162-023-00257-4/FIGURES/19
21. Chaudhuri, A.: Predictive Maintenance for Industrial IoT of Vehicle Fleets Using Hierarchical Modified Fuzzy Support Vector Machine (2018)
22. Wang, Q., Liu, J., Wei, B., Chen, W., Xu, S.: Investigating the construction, training, and verification methods of k-means clustering fault recognition model for rotating machinery. IEEE Access **8**, 196515–196528 (2020). https://doi.org/10.1109/ACCESS.2020.3028146
23. Udo, W., Muhammad, Y.: Data-driven predictive maintenance of wind turbine based on SCADA data. IEEE Access **9**, 162370–162388 (2021). https://doi.org/10.1109/ACCESS.2021.3132684
24. Ahmad, B., Mishra, B.K., Ghufran, M., Pervez, Z., Ramzan, N.: Intelligent predictive maintenance model for rolling components of a machine based on speed and vibration. In: 3rd International Conference on Artificial Intelligence in Information and Communication (ICAIIC 2021), pp. 459–464. Institute of Electrical and Electronics Engineers Inc. (2021). https://doi.org/10.1109/ICAIIC51459.2021.9415249

25. Trivedi, S., Bhola, S., Talegaonkar, A., Gaur, P., Sharma, S.: Predictive maintenance of air conditioning systems using supervised machine learning. In: 2019 20th International Conference on Intelligent System Application to Power Systems (ISAP 2019). Institute of Electrical and Electronics Engineers Inc. (2019). https://doi.org/10.1109/ISAP48318.2019.9065995
26. Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., Li, G.P.: Predictive maintenance in the Industry 4.0: a systematic literature review. Comput. Ind. Eng. **150**, 106889 (2020). https://doi.org/10.1016/j.cie.2020.106889
27. Frank, A., Asuncion, A.: {UCI} Machine Learning Repository (2010). https://archive.ics.uci.edu/ml/index.php
28. Matzka, S.: UCI Machine Learning Repository: AI4I 2020 Predictive Maintenance Dataset Data Set. https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset. Accessed 23 Jan 2022
29. Imbalanced-Learn Documentation — Version 0.9.0. https://imbalanced-learn.org/stable/. Accessed 20 Apr 2022
30. sklearn.preprocessing.MinMaxScaler — scikit-learn 1.1.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. Accessed 08 Sept 2022
31. scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation. https://scikit-learn.org/stable/index.html. Accessed 20 Apr 2022
32. sklearn.impute.KNNImputer — scikit-learn 1.1.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html. Accessed 08 Sept 2022
33. scipy.stats.zscore — SciPy v1.9.2 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html#scipy-stats-zscore. Accessed 11 Oct 2022
34. SciPy documentation — SciPy v1.9.2 Manual. https://docs.scipy.org/doc/scipy/index.html. Accessed 11 Oct 2022
35. sklearn.model_selection.GridSearchCV — scikit-learn 1.0.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. Accessed 20 Apr 2022
36. tf.keras.wrappers.scikit_learn.KerasClassifier | TensorFlow. http://man.hubwiz.com/docset/TensorFlow.docset/Contents/Resources/Documents/api_docs/python/tf/keras/wrappers/scikit_learn/KerasClassifier.html. Accessed 20 Apr 2022
37. Chollet, F., Keras, O.: The Python deep learning API. https://keras.io/. Accessed 25 Jan 2022
38. sklearn.model_selection.RandomizedSearchCV — scikit-learn 1.0.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html. Accessed 20 Apr 2022
39. sklearn.ensemble.RandomForestClassifier — scikit-learn 1.0.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed 20 Apr 2022
40. Matzka, S.: Explainable artificial intelligence for predictive maintenance applications. In: Proceedings of the 2020 3rd International Conference on Artificial Intelligence for Industries (AI4I 2020), pp. 69–74. Institute of Electrical and Electronics Engineers Inc. (2020). https://doi.org/10.1109/AI4I49448.2020.00023