

# A Comparison of State-of-the-Art Machine Learning Algorithms on Fault Indication and Remaining Useful Life Determination by Telemetry Data

Aras Firat Ünal\*  
Bilkent University  
aras.unal@ug.bilkent.edu.tr

Ali Yüce Kaleli\*  
Bilkent University  
yuce.kaleli@ug.bilkent.edu.tr

Emre Ummak  
TEKNOPAR Industrial Automation  
ummak@teknopar.com.tr

Özlem Albayrak  
TEKNOPAR Industrial Automation  
albayrak@teknopar.com.tr

**Abstract**—Contemporary trends in the diffusion of artificial intelligence technologies has increased the number of studies on predictive maintenance, a recent focus of interest in many industrial domains. Despite the increased interest in the use of machine learning for predictive maintenance, few studies involve thorough comparisons of machine learning algorithms' performance on predictive maintenance applications. This work aims to predict the remaining useful life and machine failures and compares five different algorithms: Random Forest, Gradient Boosted Tree, K-Nearest Neighbors, Multilayer Perceptron and LightGBM. Our results suggest better performances for binary classification using Random Forest, and for regression using LightGBM compared to other selected algorithms.

**Index Terms**—Failure Detection, Remaining Useful Life determination, Artificial intelligence, Predictive Maintenance, Linear regression

## I. INTRODUCTION

In the production industry, maintenance has always been an important challenge related to cost minimization, prolonged machine life, increased safety and reliability. Predictive maintenance (PdM) is one of the three maintenance strategies, in which the main goal is to estimate Remaining Useful Life (RUL) of a machinery. PdM has become an important area with the emergence of industry 4.0, and wide utilization of machine learning techniques [1]. Unplanned failures of production machinery often result in economic losses, hence, predicting machine requirements for maintenance can result in financial benefits [2] and reduce the maintenance necessiated by purely preventive approaches [3].

Optimization of maintenance scheduling has become popular due to recent developments in Artificial Intelligence (AI). AI methods have transformed maintenance scheduling from periodic maintenance, known as Preventive Maintenance, into failure indication-based maintenance PdM [4]. In the past, the paradigm in industrial maintenance focused on maintaining machinery's healthy status in large industries [5]. Maintenance was scheduled to decrease the critical failures that cause a

shorter RUL of a machinery [6]. However, in recent decades, due to its benefits in the production industry, PdM has become a widely used field for AI [7]–[10]. The motivation of this study is to compare selected machine learning algorithms for use in RUL calculation, and thus to provide basics for the stakeholders in the production industries planning to adapt the rapidly growing PdM approach. By conducting such a comparison, the study also aims to provide basics for the stakeholders in the production industries planning to adapt the rapidly growing PdM approach. The study first implements the five selected AI algorithms, namely Random Forest (RF) [11], [12], Gradient Boosted Tree (GBT) [13], [14], Light Gradient Boosting Machine (LightGBM) [15], [16], K-Nearest Neighbors (KNN) [17], [18], Multilayer Perceptron neural networks (MLP) [19], [20] and then compares different attributes of those algorithms. The study determines the RUL and detects anomalies for a specific machinery provided in the Microsoft AMLWorkshop dataset<sup>1</sup>. The dataset includes sensor data, including the values for pressure, vibration, rotation and voltage, retrieved within a year, as well as the failure occurrences for every item of machinery. The RUL of the machinery is predicted via a user-generated label. Based on the compared metrics of the implemented algorithms, the study presents the best performing AI algorithm used to estimate the RUL.

The study is organized as follows: Section II presents Related Work on the use of selected algorithms in PdM for RUL estimation. Section III provides information on the algorithms compared in the study. Section IV explains the proposed approach. Section V presents the results of the experiments. Finally, Section VI presents the conclusion.

## II. RELATED WORK

In parallel to the progress in the AI field, PdM attracted considerable research attention. This section focuses on the algorithms selected and compared in this study, and their

\* Both authors contributed equally.

<sup>1</sup><https://github.com/microsoft/AMLWorkshop/tree/master/Data>

applications to PdM tasks. Some popular work on the selected algorithms are briefly introduced in the Background section.

The work in [11] and [12] focuses on the application of RF algorithm. The PdM task in [11] is performed on the semiconductor field, where the lifespan of an ion implantation tool is determined. The problem considered in [11] is a RUL estimation problem, which is classified as a regression task. In a similar fashion, the work in [12] focuses on the PdM of wind turbines. However, rather than lifespan estimation, the work focuses on online failure indication using big data with an approach based on a machine learning for real-time PdM using big data frameworks.

PdM tasks have been fulfilled via another AI technique called Gradient Boosted Trees (GBT) in [13], [14]. In [13], the authors conducted a classification task on scheduling maintenance for railway systems. The authors of the study [14] performed RUL prediction for airline maintenance using GBT, based on internet of things with an RF algorithm as a baseline. Both [13] and [14] suggest GBTs as an alternative method to the classical RF algorithm.

Light Gradient Boosting Machine (LightGBM) is another decision tree-based algorithm that has successful applications on PdM field. For example, in [15] a combined model of convolutional neural networks and LightGBM is developed in order to forecast the status of wind turbines and their power generation. Another LightGBM application can be seen in [16], in which an improved LightGBM model is introduced to address its performance issues. The model is trained and tested on wind turbine gearbox failure detection problems.

K-nearest Neighbors (KNN), are also implemented in PdM tasks, despite being one of the most basic data-driven machine learning algorithms. For example, in [17] a methodology that can deal with the multi-dimensionality of a semiconductor manufacturing data is presented. In another example [18] Support Vector Machine (SVM) based model is supported with KNN to decrease false negative rate retrieved by the model.

The PdM literature also includes neural network based algorithms within it. There are both artificial neural network and deep learning approaches to the PdM tasks. Among the artificial neural network algorithms, the multilayer perceptron neural network (MLP) model has a wide range of applications. The work in [19] and [20] shows an MLP implementation for fault diagnosis followed by an improvement based on genetic algorithms, and a deflection classification within electrical components respectively. Among the many LSTM based approaches proposed, the work in [21] compared the vanilla LSTM model with other state-of-the-art models, such as SVM, deep Convolutional Neural Networks (CNN) and Kalman filter. Authors concluded that the proposed method achieved smaller prediction errors. As another example, the work in [22] showed that LSTM provides good diagnosis and prediction performances for many complicated cases with complex operations, hybrid faults and strong noises.

### III. BACKGROUND

In this section, the algorithms implemented throughout our study are described in order to provide a deeper understanding of the algorithms.

#### A. Random Forest

Random Forest (RF) is a machine learning algorithm consisting of many decision tree structures. Decision Trees are flowchart-like structures. It forms a tree structure with the knots and branches it hosts. Nodes mean tests for the entered attribute, while branches mean results. On the other hand, RF creates a forest structure by using these decision trees in a determined number and performs the categorization process based on the median and average values of the results. Thanks to this forest structure, it reduces the over-fitting problem that machine learning often encounters, simply memorizing the test data set and thus generates a more accurate result compared to a single decision tree. Moreover, it is versatile enough to be applied to large-scale problems, is easily adapted to various learning tasks, and can be used in both classification and regression problems [23]. The equation below describes the decision process of the algorithm where  $\hat{f}$  stands for final prediction for the input  $x'$ ,  $f_b$  stands for the output from a single tree structure and  $B$  stands for the number of baggings:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (1)$$

#### B. Gradient Boosted Trees

GBT is another modified version of a decision tree algorithm, combining decision trees to form a more concise model for the given task. The iterative procedure results in a model suitable for both regression and classification tasks. Structurally, it is effective with both numerical values and categorical values. It provides a flexible use with various error functions (loss function) and hyper parameter values. The equations below show the prediction calculations for the gradient boosted trees where  $F_m$  stands for model at  $m^{th}$  step,  $\gamma_{jm}$  is the optimal step length for the corresponding step,  $R_{jm}$  is an input space partition,  $x_i$  is the input and  $L$  is the loss function [24]:

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} \mathbf{1}_{R_{jm}}(x) \quad (2)$$

$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \quad (3)$$

#### C. Light Gradient Boosting Machine

LightGBM, an algorithm with modifications on decision trees introduced in [25], was invented to address some issues affecting decision tree algorithms. In contrast with the other tree-based algorithms, the LightGBM learns in terms of progressing vertically. Instead of increasing the numbers of levels the algorithm increases the number of leaves within its levels. The vertical growth enables faster results, and thus,

a decrease in computation power. Some popular LightGBM applications are studied in [26] and [27] focusing respectively on click fraud detection and breast cancer classification based on miRNA.

#### D. *k*-Nearest Neighbors

The algorithm KNN is created based on the concept of similar instances being close to each other in the feature space. The labels of unclassified instances are determined by consideration of a user-defined number of instances close to the unclassified instance. The algorithm is perfectly suited to classification tasks, and can be applied to regression tasks in a similar fashion [28]. The equation below shows the Euler’s distance calculation between the ground truth ( $q$ ) and the prediction ( $p$ ). In the equation,  $q_i$  is for ground truth’s and  $p_i$  is for prediction’s value for  $i^{\text{th}}$  dimension,  $n$  is the total number of dimensions in feature space. The formula is used for the determination of nearest neighbors and the labels of given instances determine the label of the unlabeled instance.

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (4)$$

#### E. Multilayer Perceptron Neural Networks

MLP neural network is a type of feed forward artificial neural network in which layers are not connected among themselves. The training of the model adjusts the weights of the connections between the nodes epoch by epoch, and the output layer is determined via the connection weights [29]. The output of the MLP neural network is determined by the activation function of the user’s choice.

### IV. PROPOSED APPROACH

Feature engineering is initially applied on the dataset to compare the algorithms in terms of accuracy, precision, recall, F1 score, mean absolute error and coefficient of determination. First of all, a single dataset was created by merging the four different datasets: ‘machines’, ‘telemetry’, ‘errors’ and ‘failures’. Then, the normalization procedure was conducted via scaling the minimum and maximum values within the dataset between 0 and 1. Following this, label column generation was applied. The regression analysis on the data set was performed by adding a linear increasing value to the data set called ‘Cycle’ as a new column. Furthermore, a linear diminishing value called ‘rul’ was also added in order to calculate the RUL. For the failure detection, a binary classification column was then used as labels in the training process for the failure detection and RUL prediction respectively. The dataset was then separated to test and training datasets, with a 1 to 3 ratio in classification, and 1 to 4 in regression. A random separation process would yield incoherent results in each training attempt, therefore, the motors with date earlier than 2015-10-01 were used for test purposes, and the rest, for the training process.

After feature engineering was applied to the datasets, the proposed models were designed. The default parameters were initially used for these models, but it was decided to apply

feature engineering to improve the performance of each model. The parameter tuning affected all algorithms. Nonetheless, the significance of this effect varied between algorithms increasing the visibility of the performance difference between the algorithms, as seen in the results section. The best set of parameters found for each model can be seen in Table I.

TABLE I  
PARAMETERS WITH THE BEST RESULTS

Algorithms	Best values
Random Forest	n_estimators=400, min_samples_split=2, min_samples_leaf=1, max_features=auto, max_depth=70, criterion=gini, bootstrap=True
Gradient Boosted Tree	n_estimators=100, learning_rate=0.1, max_features=auto', max_depth=2, criterion=friedman_mse, min_samples_split=4, min_samples_leaf=4
k-Nearest Neighbors	algorithm=auto leaf_size=25 n_neighbors=10 metric=minkowski weights=distance
Multilayer Perceptron	solver=adam, alpha=0.0001, hidden_layer_sizes=100, activation=relu, learning_rate=constant
LightGBM	lambda_T1=0.0 num_leaves=31 max_depth=-1

### V. RESULT/EXPERIMENT

The experiments were conducted on the AMLWork- shop dataset. The classification and regression tasks were conducted separately, which induces different training and testing for the tasks. The results gathered from the algorithms were then assessed through a variety of metrics, and the results retrieved from the RUL determination are visualized via the predicted RUL graphs of the first machine. The results and their visualization are presented in this section.

#### A. Data

1) *AMLWorkshop dataset:* 1) 1) The AML-Workshop dataset was considered appropriate for generalizing the findings of the experiment and seeking its implementation into Big Data. This consists of a set of sensor data i.e. voltage, pressure, vibration and rotation for 100 different machines, as well as records of all instances of errors, corresponding to five different types.

The dataset was a time-series data where the samples are the hourly sensor data gathered during 2015, resulting 876100 samples. There were nine feature columns for the dataset (four from the sensors, five from the error counts within last 24 hours, the sum of occurrences of the error within a window

length of 24 hours). For the RUL prediction, the highest cycle value was labeled as the failure point, and for the others, the RUL was determined via subtracting the machine cycle from the failure point. In contrast, for the binary classification problem, the samples of the previous 24 hours were labeled as True, and the remaining data, as False. As a result, the labels for both the regression and the classification problems were generated.

Following this, the dataset was split into training data and test data. In order to avoid the incoherence that may occur with the randomized machines, the first 75 machines were used for training, and the last 25, for testing. The final dataset, that was driven into the algorithms for training, consisted of 652075 instances, and thus the test dataset contained 224025 samples.

## B. Metrics

Suitable performance metrics for binary classification and linear regression were used to compare the models' performances. For the classification problem, the following metrics were used: accuracy, precision, recall, F1 score are used for evaluating and for the regression problem; Mean Absolute Error (MAE) and Coefficient of Determination ( $R^2$ ). In addition, training time was also computed to evaluate the models' suitability for Big Data.

1) *Binary Classification Metrics*: The evaluation of the models for the classification problem is based on the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) predictions. TP predictions are the instances in which the model correctly indicates an error, and TN predictions, those in which the model correctly indicates no failure. FP predictions are those in which the model indicates a non-existing error, and FN predictions, those in which the model overlooks a failure.

a) *Accuracy*: Accuracy is the metric for the overall success of the algorithm. It measures the ratio of True predictions to all of the predictions. The primary focus of the training is to increase accuracy, while minimizing the loss function. The formula for accuracy is the following:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

b) *Precision*: Precision is the metric that computes the proportion of correct failure indications compared to the total failure predictions. Greater precision implies that the model generates fewer false alarms. To minimize maintenance frequency, a higher precision is mandatory for the models. The precision is calculated as following:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

c) *Recall*: Recall, sometimes referred as sensitivity, is the measure for the assessment of the success of the model for indicating errors. Compared to models with a lower recall score, a model with higher recall has more success in indicating the occurrence of a failure, and misses failures less frequently. The recall is calculated via the following equation:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

d) *F<sub>1</sub> Score*: Generally, there is a trade-off between precision and recall for a single model. Tuning some parameters increases precision, but the recall may decrease. Similarly, some adjustments increase the recall score but they generally decrease the precision. Thus,  $F_1$  score, the harmonically average of recall and precision, helps evaluating the model more rigorously. The calculation for  $F_1$  Score is as follows:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (8)$$

e) *Area Under ROC Curve (AUC)*: ROC curve is the curve where the horizontal axis represents the False Positive Rate (FPR) and the vertical axis shows the True Positive Rate (TPR). The metric AUC is defined as the area that is under the ROC curve, and is a measure of the total sensitivity of the algorithm with respect to its false alarm rate.

2) *Regression Metrics*: To evaluate models for regression problems, the difference between predicted and actual value is crucial, and is the information most frequently used for the metrics. Most regression metrics are defined based on arithmetic operations on the difference between predicted and actual values. The two regression metrics used for assessing the models are Mean Absolute Error (MAE) and Coefficient of Determination ( $R^2$ ), in addition to the average training time, which is used in a similar way to a classification problem, i.e., serving the assessment of models' applicability on Big Data.

a) *Mean Absolute Error (MAE)*: The metric Mean Absolute Error (MAE) is the arithmetic average of the absolute differences between the predicted and actual Remaining Useful Life (RUL). The problem for cancellation of positive or negative errors is solved by taking the absolute value of the differences, i.e. errors. The formula for MAE calculation is as following:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

In the equation above  $n$  refers to the number of samples,  $y_i$  refers to the actual RUL for  $i^{\text{th}}$  sample and  $\hat{y}_i$  refers to the predicted value of the RUL for  $i^{\text{th}}$  sample.

b) *Coefficient of Determination ( $R^2$ )*: Coefficient of Determination ( $R^2$ ) is the proportion of variance explained by the model to the total variance of the set. The closer  $R^2$  to 1, the better the model performs. Negative  $R^2$  implies the model performs so poorly that a horizontal line explains the variation of the dataset more successfully than the model. The  $R^2$  for a single model is calculated by the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (10)$$

In the equation above  $n$  refers to the number of samples,  $y_i$  refers to the actual RUL for  $i^{\text{th}}$  sample,  $\hat{y}_i$  refers to the predicted value of the RUL for  $i^{\text{th}}$  sample and  $\bar{y}$  refers to the mean of actual values of RUL.

## C. Results

This section presents the evaluation of the performances of the proposed models. Both the classification problem and

regression problem have been evaluated based on the previously mentioned metrics and the proposed RUL prediction and anomaly detection method. Multiple tables and graphics are used to show the acquired results to allow visualization and comparison of the performances.

Table II and Table III summarize the performances of each model based on the classification metrics when parameters are tuned. On the basis of these results, it can be said that each model acquired successful results in terms of the performance metrics. It can also be said that random forests performed better compared to other models. It is also evident from the results that KNN is the one with the lowest performance since it has a low recall value that affects its overall performance. Although the performances of random forests and gradient boosted trees are similar, random forests performed slightly better. Such difference may not have a considerable impact on many cases. However, considering its usage on large-scaled projects, any incorrect forecasting would yield undesirable consequences which is the reason that this paper focused on the comparison of closely related algorithms and their parameter tuning even if the overall performances change by small amounts.

TABLE II  
BINARY CLASSIFICATION MODEL RESULTS - I

Algorithms	Metrics				
	Accuracy	Precision	Recall	F1-Score	AUC
RF	0.9993	0.9969	0.9660	0.9812	0.9830
KNN	0.9907	0.9112	0.5462	0.6830	0.7726
GBT	0.9983	0.9582	0.9483	0.9532	0.9737
MLP	0.9971	0.9482	0.8928	0.9197	0.9460
LightGBM	0.9986	0.9891	0.9357	0.9616	0.967

TABLE III  
BINARY CLASSIFICATION MODEL RESULTS - II

Algorithms	Metrics				
	TN	FN	TP	FP	Total Error
RF	72543	46	4	1307	50
KNN	72475	614	72	739	686
GBT	72491	70	56	1283	126
MLP	72481	145	66	1208	211
LightGBM	72533	87	14	1266	101

These results can also be interpreted in the context of Big Data applications, since the dataset used throughout the experiments is relatively larger than other popular datasets.

Furthermore, the results for the regression problem have also been presented in Table IV. Machine 46 data has been used for regression. From these results, it can be seen that the models performed similarly in the classification problem, but not in the RUL estimation problem. The classification problem is simpler and easier in terms of the acquisition of high performances, therefore, although the improvement from the parameter tuning was only a fraction of a percent, this considerably contributed to the performance of the models for the regression task. Hence, these results also includes insight into the sensitivity of the algorithms to parameter tuning.

Considering the presented results, LGBM algorithm acquired the best results, while GBT and RF algorithms are the ones that follow LGBM.

TABLE IV  
LINEAR REGRESSION MODEL RESULTS

Algorithms	Metrics	
	R2	Mean Absolute Error
RF	0.8157	141,177
GBT	0.8351	174,232
LightGBM	0.8832	11,591

Figures 1, 2, and 3 demonstrate the RUL predictions together with actual RUL of the models for the Motor with ID number 46. It can be observed from these figures that they are in line with our assertions about the best models. The ones that perform worse have larger fluctuations while ones with high performances almost tracked the true RUL values.

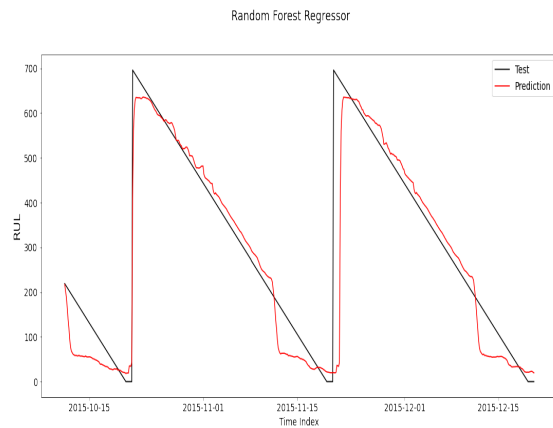


Fig. 1. RUL estimations of the Machine46 for RF model

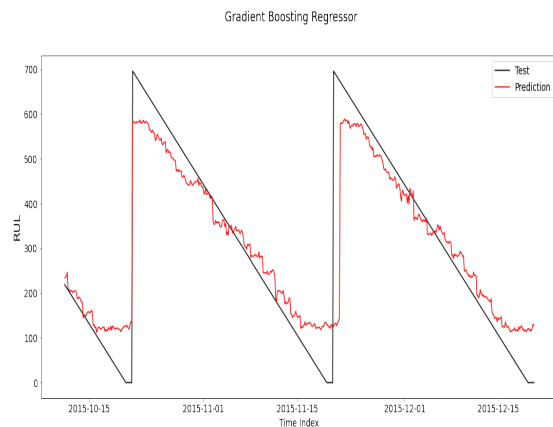


Fig. 2. RUL estimations of the Machine46 for GBT model

## VI. CONCLUSION

New technologies and in particular, developments in operations technology and information technology are increasing

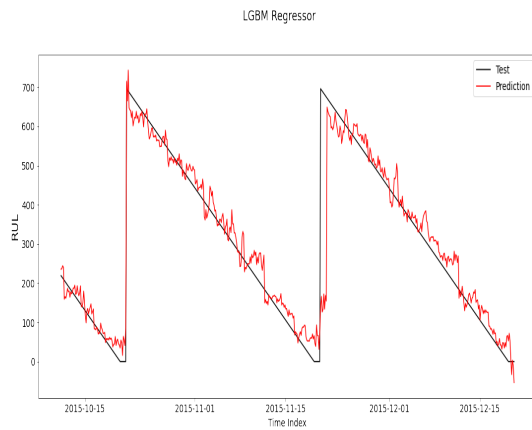


Fig. 3. RUL estimations of the Machine46 for Light GBM model

the automation of maintenance, conceptualized as PdM. This concept uses learning algorithms in order to exploit large data (e.g., distributed data) for some fundamental operations such as performance monitoring, performance evaluation, prediction of the health status, and decision-making support. As a key concept for the industry, detailed investigations in this area could further optimize the prediction performances. This paper studies the performance of different learning models on PdM, and presents a detailed comparison of those models on a recent PdM dataset.

Throughout our experiments, we utilized models with different parameters for each algorithm, tuned their parameters for best results, and applied feature engineering to generate a machine-ready input from the AMLWorkshop dataset, chosen to demonstrate the applicability of the PdM tasks to large scale problems using Big Data. This approach was used to explore several state-of-the-art learning algorithms in PdM tasks. The analysis was performed for two main problems of PdM: fault indication and RUL estimation, which correspond to classification and regression tasks respectively.

Our study shows that all the proposed algorithms are suitable for the given problems, but that some yield better performance than others. Our experiments conducted on AML-Workshop dataset identified the LightGBM as the best running model for linear regression, and RF as the best one for the binary classification. Although the dataset is highly suitable for PdM purposes, the results should not be considered to reflect the generalized performance of the given algorithms on other datasets.

Future studies should focus on other multiple datasets or problem-specific algorithms, as alternatives to the utilized dataset, and the selected algorithms.

#### ACKNOWLEDGMENT

This study has been performed by TEKNOPAR and supported partially by the COGNITWIN (Cognitive Plants Through Proactive Self-Learning Hybrid Digital Twins) project and by TÜBİTAK (The Scientific and Technological Research Council of Turkey). The COG-

NITWIN has been funded by the European Union's Horizon 2020 research and innovation programme under GA No.870130.

#### REFERENCES

- [1] C. Krupitzer, T. Wagenhals, M. Züfle, V. Lesch, D. Schäfer, A. Mozafarin, J. Edinger, C. Becker, and S. Kounev, "A survey on predictive maintenance for industry 4.0," *arXiv preprint arXiv:2002.08224*, 2020.
- [2] S.-j. Wu, N. Gebraeel, M. A. Lawley, and Y. Yih, "A neural network integrated decision support system for condition-based optimal predictive maintenance policy," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 2, pp. 226–236, 2007.
- [3] R. Baidya and S. K. Ghosh, "Model for a predictive maintenance system effectiveness using the analytical hierarchy process as analytical tool," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1463–1468, 2015.
- [4] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," *arXiv preprint arXiv:1912.07383*, 2019.
- [5] M. Dong and Y. Peng, "Equipment phm using non-stationary segmental hidden semi-markov model," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 3, pp. 581–590, 2011.
- [6] B. D. Youn, C. Hu, and P. Wang, "Resilience-driven system design of complex engineered systems," *Journal of Mechanical Design*, vol. 133, no. 10, 2011.
- [7] M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski, "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)*. IEEE, 2018, pp. 1–6.
- [8] T. P. Carvalho, F. A. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.
- [9] Ö. Albayrak and P. Ünal, "Smart steel pipe production plant via cognitive digital twins: A case study on digitalization of spiral welded pipe machinery," in *Cybersecurity workshop by European Steel Technology Platform*. Springer, 2020, pp. 132–143.
- [10] P. Unal, O. Albayrak, M. Jomaa, and A. J. Berre, "Data-driven artificial intelligence and predictive analytics for the maintenance of industrial machinery with hybrid and cognitive digital twins," in *Technologies and Applications for Big Data Value*. Springer, 2021.
- [11] P. Scheibelhofer, D. Gleispach, G. Hayderer, and E. Stadlober, "A methodology for predictive maintenance in semiconductor manufacturing," *Austrian Journal of Statistics*, vol. 41, no. 3, pp. 161–173, 2012.
- [12] M. Canizo, E. Onieva, A. Conde, S. Charramendieta, and S. Trujillo, "Real-time predictive maintenance for wind turbines using big data frameworks," in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2017, pp. 70–77.
- [13] Z. A. Bukhsh, A. Saeed, I. Stipanovic, and A. G. Doree, "Predictive maintenance using tree-based classification techniques: A case of railway switches," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 35–54, 2019.
- [14] S. Behera, A. Choubey, C. S. Kanani, Y. S. Patel, R. Misra, and A. Sillitti, "Ensemble trees learning based improved predictive maintenance using iiot for turbofan engines," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 842–850.
- [15] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, "A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting," *IEEE Access*, vol. 7, pp. 28 309–28 318, 2019.
- [16] M. Tang, Q. Zhao, S. X. Ding, H. Wu, L. Li, W. Long, and B. Huang, "An improved lightgbm algorithm for online fault detection of wind turbine gearboxes," *Energies*, vol. 13, no. 4, p. 807, 2020.
- [17] 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, 2014.
- [18] A. B. Andre, E. Beltrame, and J. Wainer, "A combination of support vector machine and k-nearest neighbors for machine fault detection," *Applied Artificial Intelligence*, vol. 27, no. 1, pp. 36–49, 2013.
- [19] M. Unal, M. Onat, M. Demetgul, and H. Kucuk, "Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network," *Measurement*, vol. 58, pp. 187–196, 2014.

- [20] I. Ullah, F. Yang, R. Khan, L. Liu, H. Yang, B. Gao, and K. Sun, "Predictive maintenance of power substation equipment by infrared thermography using a machine-learning approach," *Energies*, vol. 10, no. 12, p. 1987, 2017.
- [21] S. Zhao, Y. Zhang, S. Wang, B. Zhou, and C. Cheng, "A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method," *Measurement*, vol. 146, pp. 279–288, 2019.
- [22] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network," in *2016 IEEE international conference on aircraft utility systems (AUS)*. IEEE, 2016, pp. 135–140.
- [23] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.
- [24] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.
- [26] E.-A. Minastireanu and G. Mesnita, "Light gbm machine learning algorithm to online click fraud detection," *J. Inform. Assur. Cybersecur*, vol. 2019, 2019.
- [27] D. Wang, Y. Zhang, and Y. Zhao, "Lightgbm: an effective mirna classification method in breast cancer patients," in *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics*, 2017, pp. 7–11.
- [28] O. Kramer, "K-nearest neighbors," in *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013, pp. 13–23.
- [29] W. H. Delashmit, M. T. Manry *et al.*, "Recent developments in multilayer perceptron neural networks," in *Proceedings of the seventh Annual Memphis Area Engineering and Science Conference, MAESC*, 2005.