



# Performance Analysis of Different Machine Learning Algorithms for Predictive Maintenance

Noor A. Mohammed\*, Osamah F. Abdulateef \*\*, Ali H. Hamad\*\*\*  
and Oday I. Abdullah\*\*\*\*

\*, \*\* Department of Automated Manufacturing Engineering , Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq

\*\*\* Department of Information and Communication Engineering, Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq

\*\*\*\* System Technologies and Engineering Design Methodology, Hamburg University of Technology, 21079 Hamburg, Germany

Corresponding author \*Email: [Noor.Ali2204@kecbu.uobaghdad.edu.iq](mailto:Noor.Ali2204@kecbu.uobaghdad.edu.iq)

\*\*Email: [drosamah@kecbu.uobaghdad.edu.iq](mailto:drosamah@kecbu.uobaghdad.edu.iq)

\*\*\*Email: [ahamad@kecbu.uobaghdad.edu.iq](mailto:ahamad@kecbu.uobaghdad.edu.iq)

\*\*\*\*Email: [oday.abdullah@tuhh.de](mailto:oday.abdullah@tuhh.de)

(Received 4 October 2023; Accepted 27 November 2023; Published 1 June 2024)

<https://doi.org/10.22153/kej.2024.11.003>

## Abstract

This research offers an extensive study of different machine learning issues for predictive maintenance applications for motor condition ranging from AC motors. The research establishes a scenario that represents the real world to identify which algorithms can be trusted to predict both the time of failure and the actual type of failure in the AC motors. The employed machine learning algorithms include the Random Forest (RFC), Support Vector Classifier (SVC), K-Nearest Neighbor (KNN), Logistic Regression (LR), and XGBoost (XGB) in our work. The assessment includes the comparison of algorithms in terms of the predictive accuracy educated with different size training data. Before the model is developed, thorough data preprocessing methods will be applied that will allow the breaking down of the model assumptions and the optimization of the performance. For preprocessing step the following two steps are made including the removal of unclear samples, label encoding used for categorical columns, and column scaling. Intriguingly, the identification of seemingly outlier data points is revisited, revealing their integral role in capturing the natural variance of the dataset and enhancing classification tasks. These identified features are observed to be pivotal contributors to predictive models. The study shows that in both algorithm failure cases and failure type identification, their performances are comparable. Regarding training time, K-Nearest Neighbors (KNN) algorithm yields the top-performing model for both datasets ( 4 sec and 3 sec) respectively, whereas Random Forest (RFC) performs the worst training time (151 sec) which belong to the binary failure prediction task and XGBoost (XGB) in multi-class failure prediction task (276 sec), which is contributed. Finally, this paper emphasizes that deciding on which machine learning model is appropriate for predictive maintenance can be quite a challenge due to the necessity to balance between accuracy and training time. The findings constitute important tipping point for those companies that aim to implement a solution for predictive maintenance with the KNN model being faster and efficient at the same time.

**Keywords:** Predictive maintenance; machine learning algorithm; Random Forest (RFC); Support Vector Classifier (SVC); K-Nearest Neighbors (KNN); Logistic Regression (LR); XGBoost (XGB)

## 1. Introduction

The digitization technology offers lots of perspectives, among which is Predictive

Maintenance (PdM). By the means of sensors one could make a difference between the factors that need adjusting, the algorithm is able to translate such imperfections into preventative measures and

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



the issue could be fixed either by machine or by a human. These elements undoubtedly exert profound influence in the MPM paradigm [1]. Predictive maintenance will have future maintenance or a preventive which comprises monitoring of system/component condition based on experience, physical principles, or by machine learning techniques. The target is to acquire the locations of the bad components and deduct them before they are broken down which results into the system being offline for a shorter duration of time. Smart, monitoring-based maintenance (in fact predictive maintenance) is seen as a sustainable and affordable solution to increase availability and the amount of time that equipment can perform in the different industry sectors. Prognostics describe the process of determining the predictable functioning of the system or its components in the future, depending on their current state of health. During the predictive maintenance method development, this model will play a crucial role, being the cornerstone. The use of predictive maintenance and prognostic models greatly extends and is seen in different kind of industry environments, for example electronics, aerospace, automotive and industrial machinery, with the number of applications growing rapidly [2]. There exist two primary classes of PdM solutions: aimed at fundamentally grounded physical concepts and learning (with machine-based comprehension methods). On one hand, solving problems by way of models requires an exclusive domain expertise, which is often not accessible. On the other hand, data based approaches are designed to learn predictive models from data that is available, which renders them applicable to wide range of problems in PdM.

Machine learning provides tools for algorithms that are able to independently handle data based on complex statistics for data analysis and forecasting. The algorithms in the neural networks should be designed to work without being programmed explicitly for a particular task; rather they should be self-learning and autonomous. These algorithms should be able to authentically acquire insights from the data provided and generate accurate predictions. By the same token, machine learning has suddenly become a practical tool not only in the computer science field, but also other professions (people begin to utilize these tools to achieve their own goals) [3].

From the maintenance perspectives, the predictive maintenance with the spotlight on machine learning that is an important driver in the industry for reaching performance and cutting the costs related to the downtime is the top tool for

raising reliability of machines. Predictive maintenance includes identification of equipment failures upfront and stopping unexplainable downtime using data analyze features from various sources such as sensors, historical maintenance records, and operation records. The machine learning application has a crucial role in the predictive maintenance and automatically reveals the mistakes in the data that is missing and defective thus making the identification of the possible problems before they can escalate much easier. The historical data can be used by machine learning models to detect some predictive behaviours in equipment like impending failure. Similarly, these models can predict the length of time that machine will work and give the commuters heads-up when equipment is getting dangerously close to its failure threshold, allowing team to fix it before that happens. Through predictive maintenance which is based on machine learning techniques, the giant saving is being achieved by way of decreasing the frequency and duration of equipment breakdowns, improving efficient performance and prolonging in-service life of equipment. In addition to this, it supports maintenance teams to choose the areas that require their attention instead of the ones that are routine like the routine maintenance tasks [4].

Nowadays, Predictive maintenance (PdM) obstacles were the focus of many studies that used machine learning techniques as a tool. An actual use case from the field of PdM that used two deep learning models was provided by Silvestrin et al. [5] that compared with more classic machine learning models, which based on feature engineering. These results illustrated that even relatively simple algorithms can do learning using a small number of examples rather than deep ordering models as a result of their small number of parameters. This outcome is a clear testament for the applicability of a simple ML algorithm along with rudimentary feature engineering techniques when data availability turns out to be a hurdle in PD tasks. Also, based on 6 study by Ashok K Pundir and others [6] researchers created an intelligent PdM framework for aerospace industry that can be used both by researchers and professionals. Through their work, they accomplish both specific functions simultaneously. First, predictive maintenance will get more attention as an ML-based model will be developed. The data showed that the Random Forest model outperformed the others, the model could correctly predict TTF by  $\pm 28$  cycles with a small difference of 28.7 cycles in particular. The authors of work by Shing-Jie Pan et.al. [7] offered a technique of developing regression model for the

super-critical ultra-pressure chemical reactor having a bearing with prediction on health. This mode is based on a criteria function referred to as Piecewise Linear Remaining Useful Life (PL-RUL), which is that estimates are made from Machine Learning Regression techniques using data from acceleration and three-phased motor current (A/M) datasets. Chaitali R. Patil et. al [8] To start, information on machine health were used to present in a uniform output. Such a method collected, stored and sent audio information to the edge devices and the cloud seamlessly and a highly efficient manner. Through this tool, sensor data was experienced to visualize the machine's condition, frequently generating reports on the machine's health. To understand when the machines were failing they would use prediction tools that would ring and inform- through notifications- key players of the industry. The proposed approach aims to achieve real-time predictive maintenance for industrial machines, aligning with the goals of Industry 4.0, which prioritize process optimization, cost reduction, and enhanced efficiency as primary drivers. Mudita Uppal et al. [9], introduced a cost-effective and user-friendly intelligent office system. Their approach is suitable for practical implementation in real office environments, offering real-time monitoring of office conditions. They developed an interactive graphical user interface (GUI)-based smartphone application for overseeing the various appliances and sensors integrated into the smart office. These appliances and sensors are interconnected with Arduino, which transmits data to a cloud server. The collected data, originating from diverse devices (e.g., air conditioning, television, printer) and sensors (temperature, humidity, fire, motion, etc.), is continuously monitored through the smartphone application and server to predict potential faults using machine learning algorithms. Two different techniques, namely "K-Nearest Neighbors (KNN)" and "Naive Bayes (NB)", were applied to the dataset. KNN outperformed NB in terms of accuracy, recall, specificity, and F1-score, achieving an impressive accuracy rate of 99.63%, while NB achieved only 92.3% accuracy. Additionally, KNN exhibited a specificity of 0.99 compared to NB's 0.83. Therefore, KNN demonstrated superior performance over NB in this context. Mounia Achouch et. al [10] conducted a study where they utilized machine learning algorithms to predict failures and estimate the remaining useful life (RUL) of a TA-48 multistage centrifugal compressor. Their main goal was to reduce system downtime by applying a predictive maintenance

strategy rooted in Industry 4.0 principles. To accomplish this, they adhered to a predictive maintenance process, encompassing data exploration and preprocessing to prepare the data for model training. They performed a thorough comparison of different prediction algorithms to choose the most appropriate one, ultimately deciding on LSTM neural networks. Additionally, they improved the model's performance by regularly updating and expanding the dataset. As a result, the deployed model enabled operators to foresee compressor failures and make proactive decisions, guaranteeing minimal system downtime. Salim Qadir and Mohammed Hussein [11] applied binary classification methods for intrusion detection, employing multiple supervised machine learning algorithms as classifiers. They assessed the effectiveness of each model by measuring various metrics such as accuracy, precision, recall, f-score, error rate, true positive rate, false positive rate, and examining the confusion matrix. Most of the current PdM studies use only one machine learning algorithm, which poses a challenge to extrapolating single-model approaches to real AC motor applications. Therefore, this paper proposes a PdM method that uses five distinct machine learning algorithms, namely "Random Forest (RFC)", "Support Vector Classifier (SVC)", "K-Nearest Neighbors (KNN)", "Logistic Regression (LR)", and "XGBoost (XGB)", for the purpose of predicting potential machine failures and conducting a comparative analysis of these algorithms. To assess the effectiveness of these machine learning techniques in a practical Predictive Maintenance (PdM) context, we employed the AC motor condition monitoring dataset. Our approach involves training these algorithms with datasets of varying sizes, and subsequently examining how changes in the size of the training data impact their performance by comparing accuracy curves on the test data. The proposed PDM can not only monitor the state of the AC motor in real-time but also predict the fault of the motor in advance.

## 2. Implementing A Machine Learning Algorithms

Machine learning is a discipline focused on enabling computers to acquire knowledge without the need for explicit programming. Arthur Samuel gained renown for his checkers-playing program, a notable early example. Machine learning (ML) serves the purpose of instructing machines on how

to effectively process data, particularly when data interpretation is challenging. ML becomes necessary when we struggle to extract meaningful insights from information. Given the wealth of available datasets, the demand for machine learning is steadily increasing, with numerous industries relying on it to derive valuable information. In essence, the primary objective of machine learning is to gain knowledge from data. Machine Learning depends on various algorithms to address data challenges. It's important to note that there isn't a universal algorithm that suits all problems; the choice of algorithm depends on factors such as the problem type, the number of variables involved, and the most suitable model. Let's briefly explore the ML algorithms employed in this research [12].

#### • Random Forest

A Random Forest (RFC) is an ensemble machine learning approach used for predictive modeling, suitable for both classification and regression tasks. It combines the outcomes from various decisions made by different decision trees. The last stage is done by the determination of the verification of the mode (most frequent value) for classification or mean prediction from these trees for regression. The process gets on by splitting the data set into the training set and test set, then selecting the entire samples randomly from the training set. Each of these categories uses a decision tree to break the divisions with the goal of accurate segmenting of data points into groups or predicting values. This is sequential firstly picking up samples and then making decisions trees. Then the Random Forest combines the individual predictions of all the decision trees and finally it conducts a majority vote. Those predictions that have the majority of votes are actually the final output values [12].

#### • Support Vector Classifier

SVC ( Support Vector Classifiers ) which is a support vector is the most widely-used system in machine learning. In the context of machine learning, Support Vector Classifiers are supervised learning models having an algorithm which analyses the data to perform the classification and regression analysis functions. Progressing from their competency to execute linear classification, SVCs has come to exhibit non-linear classification through the procedure of “the kernel trick.” This method produces the necessary k-dimensional feature spaces that mark particular classes based on typical features associated with each class. While the design is complex and sensitive to the balance between accuracy and coverage, each boundary is passed through with the aim of minimizing the

distance between them and the separate classes to prevent misclassification errors [13].

#### • K-Nearest Neighbor

The K-nearest neighbors method (KNN) belongs to the supervised learning category that is, into the class of classification. KNN stands for the nearest neighbors algorithm that uses all data available and is insensitive to noise by evaluating similarity. Through a measurement of two parameters in a two dimensional Cartesian plane, we can establish the similarity of the points on the plane based on the distance between them. On the same wavelength, KNN works based on the law that the similar things usually stay close to each other.[14]

#### • Logistic Regression

Logistic regression (LR), being a statistical method that uses logistic function as its core unit. Though there are advanced derivations of this model, its ultimate usage is to calculate the probability of either a binary dependent variable. The logistic regression model itself doesn't carry out statistical classification – it's not a classifier per se. However, it can be adapted for classification tasks. One common technique is to establish a threshold value and categorize inputs based on whether their predicted probabilities surpass or fall below this threshold. This method is widely adopted for constructing binary classifiers. Logistic regression finds applications in various domains such as fraud detection, clinical trials, and scenarios where outcomes are binary [13].

#### • XGBoost

XGBoost (XGB) is a highly efficient gradient tree boosting technique that creates a sequence of decision trees. It excels at swiftly carrying out relevant computations across various computational setups. Consequently, XGBoost is widely favored for its capability to model new features and make classifications. The utilization of the XGBoost algorithm has surged in popularity, mainly owing to its integration with tabular and structured datasets [15].

### 2.1 Gathering data

As previously mentioned, machine learning algorithms undergo training using existing datasets to enhance their performance. The effectiveness of these models relies significantly on the size and quality of the training data. The initial phase of the machine learning workflow involves identifying, collecting, or in some instances, generating the training dataset. The specific approach taken in this regard is heavily influenced by the desired

objectives of the model under development. To gain practical experience in executing the various stages of a machine learning workflow, we employed datasets obtained from sensors monitoring a universal AC motor. These datasets encompass normal operating conditions, fault

scenarios, and include a motor with bidirectional rotation capability as well as a rapid motor stop feature. Table 1 shows the size and counts of data sets for each classification, while Table 2 shows samples of the data collected.

**Table 1,**  
**Size and counts of data sets for each classification [16]**

Failure Type	File Size (KB)	Count of Data
No failure	654.3	11996
vibration failure	316.1	5946
over current failure	550.7	10092
Bush failure	157.7	2910
Stop rotating failure	254.6	4656

**Table 2,**  
**Sample of data collected [16]**

	accel x (1g)	accel y (1g)	accel z (1g)	amb_temp (°C)	Object_temp (°C)	Current (Amp.)	Failure type
1	3.656	-1.125	-7.156	22.35	51.69	0.586	No Failure
2	0.593	-0.625	-7.187	22.38	51.53	0.587	No Failure
3	-2.5	0.593	-6.687	22.36	51.61	0.610	No Failure
4	2.781	-1.281	-6.812	22.36	51.63	0.519	No Failure
5	2.156	-0.687	-7.343	22.38	51.49	0.558	No Failure
6	-2.25	1.062	-6.937	22.36	51.67	0.606	No Failure
7	-1.906	0.687	-1.406	22.88	53.03	0.569	Vibration Failure
8	4.781	-1.562	-8.031	23.15	55.35	1.339	Over Current Failure
9	0.718	-0.406	-6.875	23.81	65.53	1.041	Bush Failure
10	-16	-16	-11.375	23.74	54.89	0.672	Stop Rotating Failure

## 2.2 Outliers inspection

The goal of this section is to check if the dataset contains any outliers, which are usually misleading for machine learning algorithms. We start by examining a statistical analysis of the numerical characteristics. It is possible to infer the potential existence of outliers in both the x-axis acceleration and object temperature due to the substantial disparity between the maximum value and the third quartile. To provide a clearer perspective on this, we delve deeper into the data using boxplots to gain

insights into the distribution, as illustrated in Figure 1, however, in the case of acceleration in x-axis, there are probably traceable outliers to the way outliers are detected using boxplots. In the case of object temperature, the Gaussian distribution is skewed and it is not unrealistic to think that the few observations with medium object temperatures are going to fail. As a result, the outliers were retained and reserved the discretion to determine whether to take any action on them after evaluating other factors.

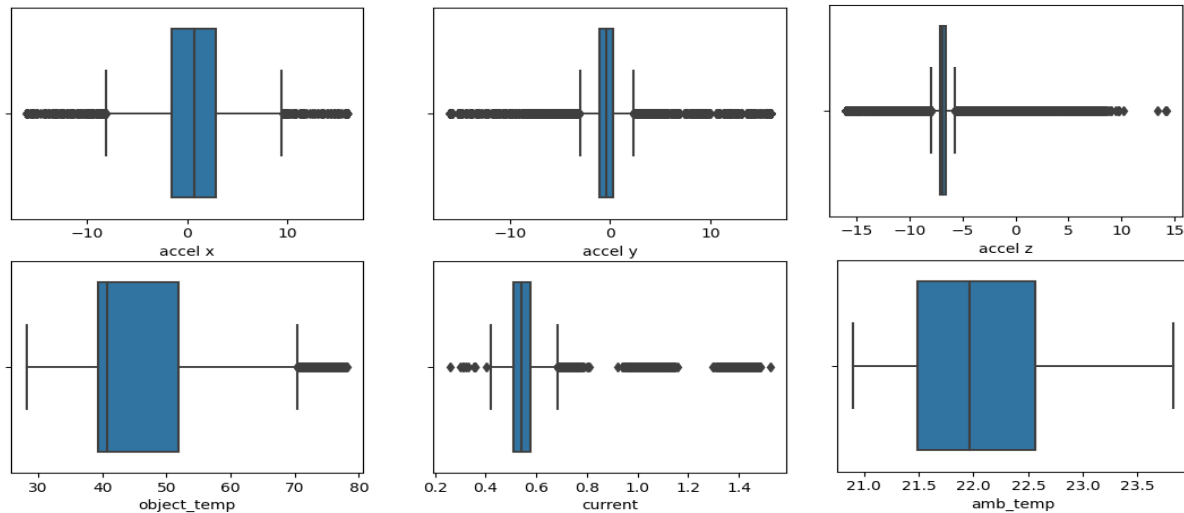


Fig. 1. Numeric features boxplots.

### 2.3 Resampling with Smote

Resampling with the Synthetic Minority Oversampling Technique (SMOTE) is an important consideration considering the extremely low occurrence of machine failures among the entire dataset. When addressing machine learning challenges, the issue of class imbalance becomes a significant concern. This imbalance can distort both the training process of models and our ability to interpret their outcomes. For instance, if we create a model using a dataset that predicts machines will never malfunction, it might seem accurate on the surface. To mitigate these issues and minimize biased model behavior toward specific classes, we employ data augmentation.

The goal is to achieve an 80-20 ratio between functioning and faulty observations, along with an equal occurrence percentage among the various failure causes. The SMOTE procedure operates as follows: It randomly selects a subset from the minority class and identifies the  $k$  nearest neighbors for each observation in this subset. Next, it selects one of these neighbors and calculates the vector connecting the current data point to the chosen neighbor. This vector is then scaled by a random value ranging from 0 to 1. The synthetic data point is generated by adding this scaled vector to the current data point. The result of resampling is shown in Figure 2.

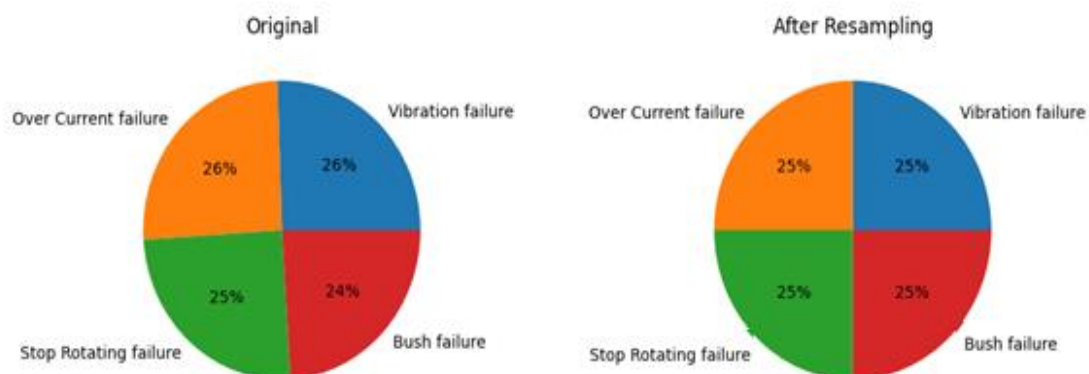


Fig. 2. Causes involved in machine failures.

## 2.4 Metrics

To evaluate the models, we used from a quantitative point of view, some metrics that summarized some characteristics of the classification results:

Accuracy: expresses the fraction of instances that are classified correctly; it is the most intuitive metric that is usually used in classification tasks [17].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \dots(1)$$

Where:

True Positive (TP): when both the actual and predicted values are 1.

True Negative (TN): when both the actual and predicted values are 0.

False Positive (FP): when the actual value is 0 but the predicted value is 1.

False Negative (FN): when the actual value is 1 but the predicted value is 0.

AUC: can be considered as a measure of the separation between True Positives and True Negatives, that is, the ability of the model to distinguish between classes.

F1: reports the classification capacity of the model to Precision and Recall, giving both the same weight. [17]

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad \dots (2)$$

Where:

$$Precision = \frac{TP}{TP+FP} \quad \dots (3)$$

$$Recall = \frac{TP}{TP+FN} \quad \dots (4)$$

Although generally effective, AUC can be optimistic in the case of highly unbalanced classes, as happens in the binary task, while the F1 score is more reliable in this kind of scenario. We consider this last metric particularly significant as it is able to mediate the cases in which the machines that are about to fail are classified as functioning (Recall) and the ones in which functioning machines are classified as about to suffer a failure (Precision). To be more specific, we gave more importance to recall than Precision by evaluating an "adjusted" version of the F1 through a  $\beta$  parameter [17]:

$$F\beta = (1 + \beta^2) \frac{Precision * Recall}{\beta^2 Precision + Recall} \quad \dots (5)$$

With the choice  $\beta=2$  (common in literature) a greater influence of the Recall is obtained. The decision to limit unnecessary replacement material purchases is driven by the goal of optimizing machinery maintenance costs. However, it is even more crucial to prevent machinery breakdowns, as they typically entail higher expenses, outweighing the savings from reducing replacement material purchases.

Figure 3 shows a flowchart that summarizes the main steps used in predicting failures in AC motors using machine learning algorithms.

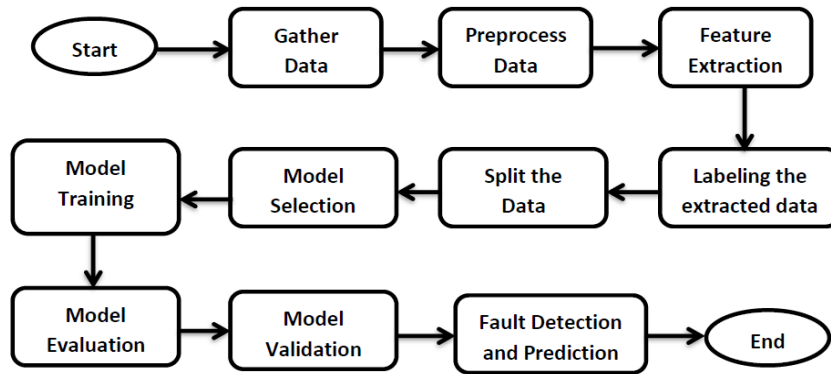


Fig. 3. Flowchart of a proposed approach.

## 3. Results and Discussion

### 3.1 Binary task

The goal of this section was to find the best model for binary classification of the dataset to predict whether or not there was machine failure. Classification algorithms, a key component of data mining, employ supervised machine learning techniques to forecast data outcomes. They operate by utilizing a pre-labeled dataset containing

multiple classes or categories, constructing a classification model based on this input, and subsequently applying it to new, unlabeled data to determine their respective class assignments. Typically, the initial dataset is partitioned into three sets: the training set, used for model training; the validation set, employed to fine-tune model parameters and assess its performance on the training data; and the test set, reserved for evaluating the model. In our study, we adopted a split ratio of 80% of each failure type count of data mentioned in Table 1 for training, 10% for



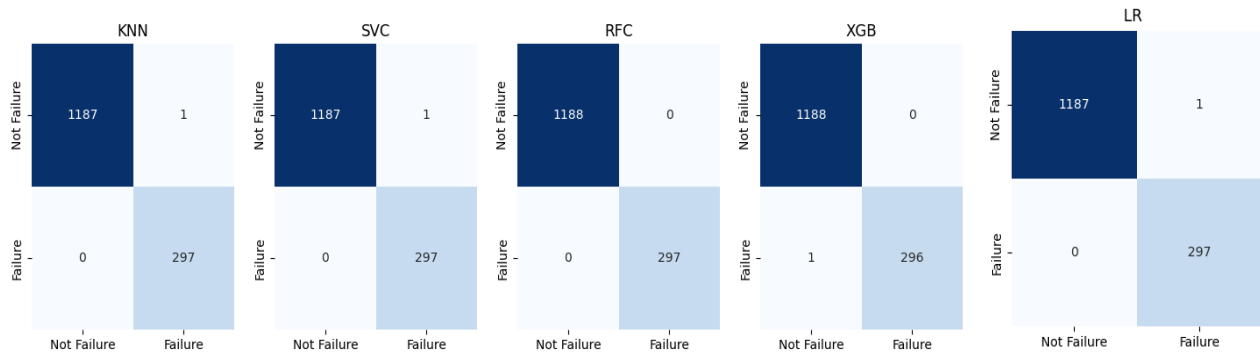
validation, and 10% for testing. Our experimentation revealed that this partitioning strategy yielded the optimal results among the various strategies we considered.

All of the chosen models achieved comparable results on the validation set, as depicted in Table 3. It is challenging to ascertain the superiority of one model over another based solely on these values. Moreover, the performance remained consistent when evaluated on the test set, as demonstrated in Table 4, indicating that overfitting was successfully mitigated. In order to fully verify the robustness of the AC motor, the state of the AC motor was predicted by examining the confusion

matrices and the metrics showcased in Figures 4 and 5. The horizontal axis is the label corresponding to the predicted state and the vertical axis is the label corresponding to the actual state. This approach helps to elucidate a hierarchical ranking among the utilized models, as the metrics for each model consistently either outperform or underperform compared to the others, while the parameter search time remains similar. All models (LR, KNN, XGB SVC, and RFC) achieved extremely similar results, but the training time is varied as shown in Table 5. In particular RFC obtains the worst training times and KNN the best ones.

**Table 3,**  
**Validation scores of a binary task**

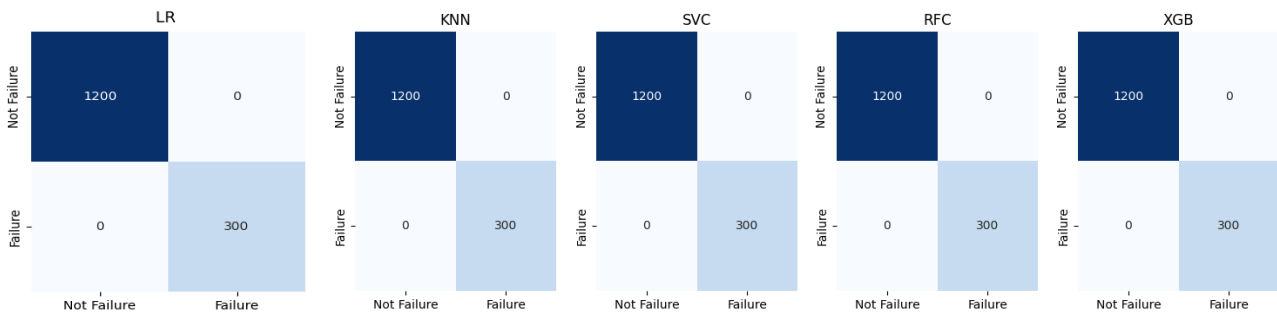
	LR	KNN	SVC	RFC	XGB
ACC	0.999	0.999	0.999	1.0	0.999
AUC	1.0	1.0	1.0	1.0	1.0
F1	0.998	0.998	0.998	1.0	0.998
F2	0.999	0.999	0.999	1.0	0.997



**Fig. 4. Validation set confusion matrices of a binary task**

**Table 4,**  
**Test scores of a binary task**

	LR	KNN	SVC	RFC	XGB
ACC	1.0	1.0	1.0	1.0	1.0
AUC	1.0	1.0	1.0	1.0	1.0
F1	1.0	1.0	1.0	1.0	1.0
F2	1.0	1.0	1.0	1.0	1.0



**Fig. 5. Test set confusion matrices of a binary task.**



**Table 5,**  
**Training time for each algorithm of a binary task**

Machine Learning Algorithms	Training time (sec)
LR	8
KNN	4
SVC	34
RFC	151
XGB	83

### 3.2 Multi-class task

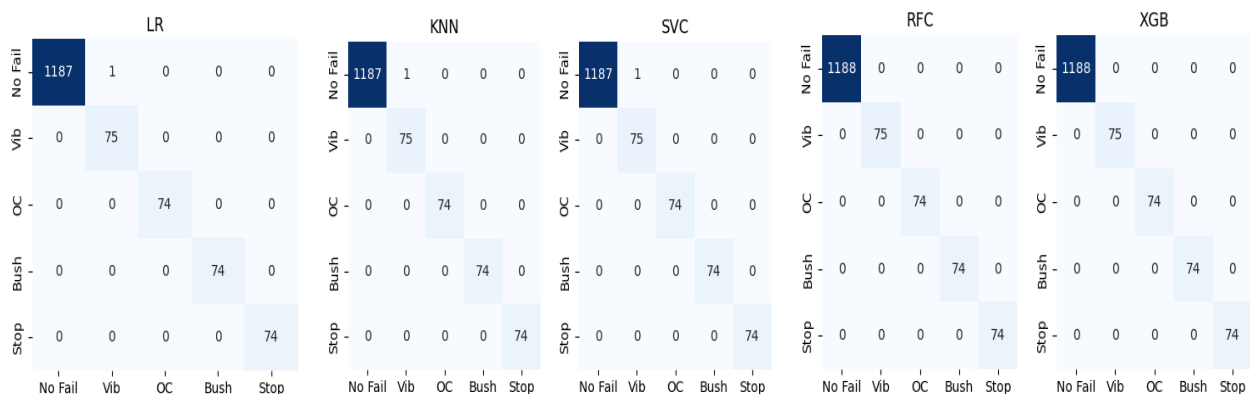
The second task of this work was to predict not only if there would be a failure but also the type of failure that occurred. So, we were dealing with multiclass classification problems, assuming that each sample was assigned to one and only one label. This hypothesis was verified because, in data preprocessing, we removed all the ambiguous observations that belonged to more than one class. For multiclass targets, when we calculated the values of AUC, F1, and F2 scores in Tables 6 and 7, we needed to set the parameter 'average.' We chose 'average=weighted' to account for class imbalance. In fact, at the end of data preprocessing, we had 80% working machines and 20% that failed.

As for the binary classification task, we adapted the models developed in the previous section. While many classification algorithms (such as 'Random K-nearest neighbor,' 'Random Forest,' and 'XGBoost') naturally permitted the use of more than two classes, some (like Logistic Regression and Support Vector Classifier) were by nature binary algorithms. However, these could be turned into multiclass classifiers using a variety of strategies. For our work, we decided to train a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. We chose this because it was computationally more efficient than other types of approaches.

For each model, we launched the grid search for hyperparameter optimization, using the weighted average F2 score as the metric to evaluate the model. By comparing the results obtained, we saw that all models obtained high values for the chosen metrics both for the validation and test sets, as shown in Figures 6 and 7. If we look at the training time, KNN took the shortest time, while XGB took a very long time, as shown in Table 8. So, one could choose the KNN model according to their needs.

**Table 6,**  
**Validation scores of a multi-class task**

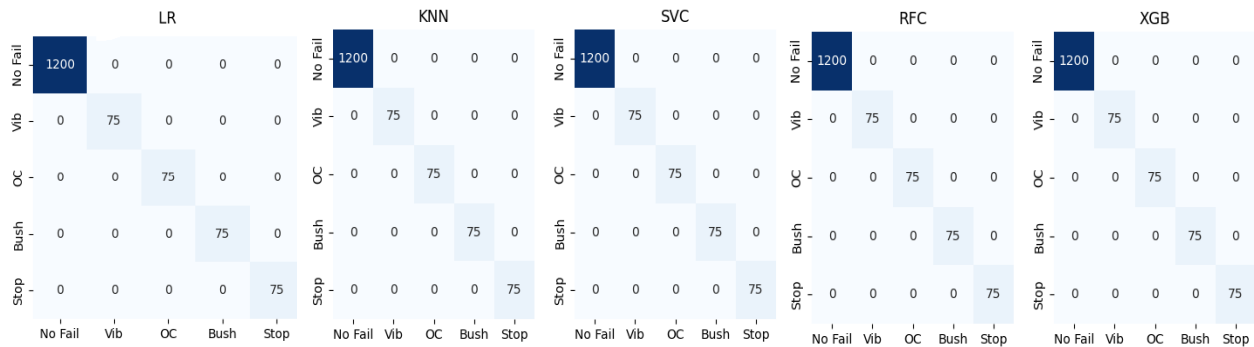
	LR	KNN	SVC	RFC	XGB
ACC	0.999	0.999	0.999	1.0	0.999
AUC	1.0	1.0	1.0	1.0	1.0
F1	0.999	0.999	0.999	1.0	1.0
F2	0.999	0.999	0.999	1.0	1.0



**Fig. 6. Validation set confusion matrices of a multi-class task.**

**Table 7,**  
**Test scores of a multi-class task**

	LR	KNN	SVC	RFC	XGB
ACC	1.0	1.0	1.0	1.0	1.0
AUC	1.0	1.0	1.0	1.0	1.0
F1	1.0	1.0	1.0	1.0	1.0
F2	1.0	1.0	1.0	1.0	1.0



**Fig. 7. Test set confusion matrices of a multi-class task.**

**Table 8,**  
**Training time for each algorithm of a multi-class task**

Machine Learning Algorithms	Training time (sec)
LR	10
KNN	3
SVC	53
RFC	139
XGB	276

### 3. Conclusions

In this research work, datasets pertaining to the condition monitoring of AC motors was utilized to assess the effectiveness of various machine learning techniques within a practical Predictive Maintenance (PdM) context. We employed five different machine learning algorithms, namely “Random Forest (RFC)”, “Support Vector Classifier (SVC)”, “K-Nearest Neighbors (KNN)”, “Logistic Regression (LR)”, and “XGBoost (XGB)”, to predict potential machine failures. Additionally, we conducted a comparative analysis of these algorithms. Our approach involves training these algorithms using datasets of varying sizes and subsequently evaluating their accuracy on test data. This evaluation was conducted in relation to the size of the training data, allowing us to investigate how the training set size impacts algorithm performance. The main contribution of this paper compared to other related works is developing a PdM method using five machine learning

algorithms that can not only monitor the state of the AC motor in real-time but also predict the fault of the AC motor in advance. The outcomes of our analyses and the resulting findings enable us to draw meaningful conclusions regarding this research project.

Two primary objectives were understood: firstly, to predict machine failure, and secondly, to forecast the specific failure type. Prior to constructing our models, we engaged in comprehensive data preprocessing to validate model applicability assumptions and optimize performance. In this preprocessing phase, we removed certain ambiguous samples, performed label encoding on categorical columns, and applied column scaling. Additionally, we initially identified some data points as outliers, which, upon further examination, were revealed to be intrinsic to the data's natural variability and played a significant role in the classification task. Consequently, we determined that these features had the most substantial impact on predictions when applying the models. Interestingly, our findings, according to the case of the AC motor, demonstrated that the machine's type did not influence the occurrence of failures. We can conclude that for both task the models are the same in their performance, and the only difference is the training time. For both tasks the best model is KNN, and the worst is SVC for binary task, and XGB for multi-class task according to the response

time. The choice of the model depends on the needs of the company; for faster application, one can use KNN.

Further, there are more challenges to be considered in future research, such as the ability to predict unknown faults, which needs to be optimized. Future research can be extended to the use federated learning between many AC motors that is dynamic in client sample sizes, client distributions, and the client set as a whole.

## Symbols and Abbreviations

<b>AUC</b>	Accuracy
<b>FN</b>	False Negative
<b>FP</b>	False Positive
<b>KNN</b>	K-Nearest Neighbors
<b>LR</b>	Logistic Regression
<b>ML</b>	Machine Learning
<b>NB</b>	Naive Bayes
<b>PdM</b>	Predictive Maintenance
<b>RFC</b>	Random Forest
<b>SMOTE</b>	Synthetic Minority Oversampling Technique
<b>SVC</b>	Support Vector Classifier
<b>TN</b>	True Negative
<b>TP</b>	True Positive
<b>TTF</b>	Time to Failure
<b>XGB</b>	XGBoost

## References

- [1] Pech, M.; Vrchota, J.; Bednář, J. Predictive Maintenance and Intelligent Sensors in Smart Factory: Review. *Sensors* 2021, 21, 1470. <https://doi.org/10.3390/s21041470>
- [2] Andreas, T.; Judith, P.; Marcel, K.; Gordon, E. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability Engineering & System Safety* 2021, 215, 107864. <https://doi.org/10.1016/j.ress.2021.107864>
- [3] Onur, S.; Stephen, A.; John, Y. Condition Monitoring using Machine Learning: A Review of Theory, Applications, and Recent Advances. *Expert Systems with Applications* 2023, 221, 119738. <https://doi.org/10.1016/j.eswa.2023.119738>
- [4] [4] Rojek, I.; Jasiulewicz Kaczmarek, M.; Piechowski, M.; Mikołajewski, D. An Artificial Intelligence Approach for Improving Maintenance to Supervise Machine Failures and Support Their Repair. *Appl. Sci.* 2023, 13, 4971. <https://doi.org/10.3390/app13084971>
- [5] Silvestrin, L.P.; Hoogendoorn, M.; Koole, G. A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance. In *Proceedings of the SSCI, Xiamen, China, 6–9 December 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 760–767. <https://doi.org/10.1109/SSCI44817.2019.9003044>
- [6] Ashok, K. P.; Pratik, M.; Pradeep, P. Machine Learning Based Predictive Maintenance Model. *Proceedings of the 2nd Indian International Conference on Industrial Engineering and Operations Management, Warangal, Telangana, India, August 2022*.
- [7] Pan, S.-J.; Tsai, M.-L.; Chen, C.-L.; Lin, P.T.; Lee, H.-Y. Investigation of Machine Learning Methods for Predictive Maintenance of the Ultra-High-Pressure Reactor in a Polyethylene-Vinyl Acetate Production Process. *Electronics* 2023, 12, 580. <https://doi.org/10.3390/electronics12030580>
- [8] Chaitali, R. P.; Sanika, K. J.; Asmeeta, L. B.; Ankita P. D.; Mahee P. R. Machine Learning-Based Predictive Maintenance of Industrial Machines. *International Journal of Computer Trends and Technology* 2023, 71, 3, 50-56. <https://doi.org/10.14445/22312803/IJCTT-V71I3P108>
- [9] Uppal, M.; Gupta, D.; Goyal, N.; Imoize, A.L.; Kumar, A.; Ojo, S.; Pani, S.K.; Kim, Y.; Choi, J. A Real-Time Data Monitoring Framework for Predictive Maintenance Based on the Internet of Things. *Complexity* 2023, 9991029. <https://doi.org/10.1155/2023/9991029>
- [10] Achouch, M.; Dimitrova, M.; Dhouib, R.; Ibrahim, H.; Adda, M.; Sattarpanah Karganroudi, S.; Ziane, K.; Aminzadeh, A. Predictive Maintenance and Fault Monitoring Enabled by Machine Learning: Experimental Analysis of a TA-48 Multistage Centrifugal Plant Compressor. *Appl. Sci.* 2023, 13, 1790. <https://doi.org/10.3390/app13031790>
- [11] Qadir Mohammed, salim; A. Hussein, M. Performance Analysis of Different Machine Learning Models for Intrusion Detection Systems. *jcoeng* 2022, 28, 61-91. <https://doi.org/10.31026/j.eng.2022.05.05>
- [12] Fayez, T. K.; Wijdan, A.; Zahra, G. Random Forest Machine Learning Technique for Automatic Vegetation Detection and Modelling in LiDAR Data. *Int J Environ Sci Nat Res.* 2021, 28, 2. <https://doi.org/10.19080/IJESNR.2021.28.556234>

- [13] Susmita R. A Quick Review of Machine Learning Algorithms. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, 35-39.  
<https://doi.org/10.1109/COMITCon.2019.8862451>
- [14] Mohsin, M.A.; Hamad, A.H. Performance evaluation of SDN DDoS attack detection and mitigation based random forest and K-nearest neighbors machine learning algorithms. *Revue d'Intelligence Artificielle* 2022, 36, 2, 233-240. <https://doi.org/10.18280/ria.360207>
- [15] Jian, F.; Hongbin, W.; Fan Y.; Kuang, Y.; Xiang, L.; Min, Z. A failure prediction method of power distribution network based on PSO and XGBoost. *Australian Journal of Electrical and Electronics Engineering* 2022, 19, 4, 371-378.  
<https://doi.org/10.1080/1448837X.2022.2072447>
- [16] Mohammed, N.A.; Abdulateef, O.F.; Hamad, A.H. An IoT and machine learning-based predictive maintenance system for electrical motors. *Journal Européen des Systèmes Automatisés* 2023, 56, 4, 651-656.  
<https://doi.org/10.18280/jesa.560414>
- [17] Jérôme, D.; Gaël, V.; Jean-Baptiste, P.; Preventing dataset shift from breaking machine-learning biomarkers. *GigaScience* 2021, 10, 9.  
<https://doi.org/10.1093/gigascience/giab055>

## تحليل أداء خوارزميات التعلم الآلي المختلفة للصيانة التنبؤية

نور علي محمد\*، أسامة فاضل عبد اللطيف\*\*، علي حسين حمد\*\*\*، عدي إبراهيم عبد الله\*\*\*\*

\*قسم هندسة التصنيع المؤتمت، كلية الهندسة الخوارزمي، جامعة بغداد، بغداد، العراق

\*\*قسم هندسة المعلومات والاتصالات، كلية الهندسة الخوارزمي، جامعة بغداد، بغداد، العراق

\*\*\*\*تقنيات النظام ومنهجية التصميم الهندسي، جامعة هامبورغ للتكنولوجيا، هامبورغ 21079، ألمانيا

\* البريد الإلكتروني: [Noor.Ali2204@kecbu.uobaghdad.edu.iq](mailto:Noor.Ali2204@kecbu.uobaghdad.edu.iq)

\*\*البريد الإلكتروني: [drosamah@kecbu.uobaghdad.edu.iq](mailto:drosamah@kecbu.uobaghdad.edu.iq)

\*\*\*البريد الإلكتروني: [ahamad@kecbu.uobaghdad.edu.iq](mailto:ahamad@kecbu.uobaghdad.edu.iq)

\*\*\*\*البريد الإلكتروني: [oday.abdullah@tuhh.de](mailto:oday.abdullah@tuhh.de)

### المستخلص

يقدم هذا العمل دراسة شاملة عن تقييم أداء خوارزميات التعلم الآلي المختلفة للصيانة التنبؤية في سياق مراقبة حالة محرك التيار المتردد. ويركز البحث على سيناريوهات واقعية لتقييم قدرات الخوارزميات المختلفة في التنبؤ في كل من حدوث الأعطال ونوع الفشل المحدد في محركات التيار المتردد. ويتم استخدام خمس خوارزميات بارزة للتعلم الآلي، وهي Random Forest (RFC)، و Support Vector Classifier (SVC)، و K-Nearest Neighbors (KNN)، والانحدار اللوجستي (LR)، و XGBoost (XGB)، في هذا العمل. يتضمن التقييم تحليلًا مقارنًا لهذه الخوارزميات بناءً على دقتها التنبؤية، مع مجموعات تدريب بأحجام مختلفة. قبل تطوير النموذج، ويتم تطبيق تقنيات معالجة مسبقة صارمة للبيانات لضمان صحة افتراضات النموذج وتحسين الأداء. وتتضمن مرحلة المعالجة المسبقة إزالة العينات الغامضة، وترميز الأعمدة الفئوية باستخدام ترميز المصنفات، وقياس الأعمدة. ومن المثير للاهتمام، إعادة النظر في تحديد نقاط البيانات التي تبدو غريبة، مما يكشف عن دورها الأساسي في النقاط التباين الطبيعي لمجموعة البيانات وتعزيز مهام التصنيف. ولوحظ أن هذه الميزات المحددة هي مساهمات محورية في النماذج التنبؤية. ويوضح العمل أنه بالنسبة للتنبؤ بفشل الآلة وتحديد أنواع الفشل، فإن الخوارزميات تظهر أداءً مشابهًا. ويكمن الاختلاف الأساسي في وقت التدريب، إذ تبرز خوارزمية K-Nearest Neighbors (KNN) باستمرار بوصفها النموذج الأفضل أداءً لكلا المهمتين (4 ثوانٍ و 3 ثوانٍ) على التوالي، بينما تؤدي Random Forest (RFC) أسوأ وقت للتدريب (151 ثانية). (في مهام التنبؤ بالفشل الثنائي، و XGBoost (XGB) في مهام التنبؤ بالفشل متعددة الفئات (276 ثانية)، والتي تعزى إلى أوقات الاستجابة الخاصة بها. في الختام، تؤكد هذه الورقة أهمية اختيار نماذج التعلم الآلي المناسبة للصيانة التنبؤية، مع الأخذ في الاعتبار المفاضلة بين الدقة ووقت التدريب. وتوفر النتائج رؤى قيمة للشركات التي ترمي إلى تنفيذ استراتيجيات صيانة تنبؤية تنسجم بالكفاية والفعالية، مع ظهور KNN كخيار مثالي للتطبيق السريع.