Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2025, 19(11), 33–44
https://doi.org/10.12913/22998624/208284
ISSN 2299-8624, License CC-BY 4.0

Received: 2025.05.26
Accepted: 2025.09.15
Published: 2025.10.01

Comparison of machine learning methods in predictive maintenance of machines

Michał Cioch[1*], Monika Kulisz[1], Arkadiusz Gola[1]

[1] Lublin University of Technology, ul. Nadbystrzycka 38d, 20-618 Lublin, Poland
* Corresponding author's e-mail: m.cioch@pollub.pl

ABSTRACT

The objective of this study is to identify the most effective machine learning algorithm for predictive maintenance of industrial machinery using three input variables: temperature, vibration, and machine condition. Considering the balance between predictive accuracy and computational efficiency, as well as the practicality of implementation in resource-constrained environments. This study evaluated the effectiveness of six machine learning algorithms for predictive maintenance in industrial environments using three input variables. A dataset of 90,000 training instances and 10.000 test instances was analyzed, with models including decision trees, neural networks, support vector machines (SVMs), k-nearest neighbor (KNN), naive Bayes, and logistic regression. Performance was evaluated based on accuracy, F1 score, AUC, training time, prediction speed, and model size. The results showed that the coarse decision tree achieved the highest accuracy (98.24%), the lowest error rate (1.76%) and the highest prediction speed (> 420,000 observations/second) with the smallest model size (4.7 KB). The results underscore that simpler, easy-to-interpret models, such as decision trees, offer excellent practicality for real-time industrial applications without compromising predictive power. This work highlights the importance of balancing model complexity with computational efficiency in predictive maintenance systems.

Keywords: predictive maintenance, machine learning, decision trees, neural networks, computational efficiency, model accuracy, industrial applications.

INTRODUCTION

Modelling of technical and organizational processes plays a key role in modern production and maintenance management systems. Not only does it provide a better understanding of the dynamics of the operation of machines and production lines, but it also makes it possible to simulate different operating scenarios and assess the impact of parameter changes on the efficiency of the overall system. Mathematical, simulation and machine learning based models are increasingly used to analyze and predict complex environmental phenomena, such as the metabolism of aquatic ecosystems, pollutant emissions or the efficiency of recycling processes [1–3]. In the literature, one finds both approaches based on deterministic and simulation models (e.g., Enterprise Dynamics) and statistical or regression methods to predict system and non-system behavior based on historical data [3–6]. The integration of modeling with advanced data analytics and machine learning opens up new possibilities for predictive maintenance, enabling not only the detection of potential failures, but also the optimization of service schedules and the minimization of operating costs [8–10].

Predictive maintenance (PdM) has become a key strategy in industrial applications to minimize unplanned downtime and operational costs. Using sensor data and machine learning (ML). PdM enables early detection of equipment failures and optimizes maintenance schedules [11]. Traditional approaches such as time-based or reactive maintenance are increasingly being replaced by data-driven solutions that analyze variables such as temperature, vibration and machine condition to predict maintenance needs [12–14].

In recent years, PdM solutions have increasingly been realized with the direct use of edge devices, such as PLCs or smart sensors [15]. This allows data to be processed and predictions to be made in real time, minimizing latency and the need to send data to the cloud. Thus, even with limited computing resources, state-of-the-art algorithms, including neural networks, are efficiently implemented on compact industrial platforms [16]. Recent research in machine learning has demonstrated the effectiveness of algorithms such as decision trees, neural networks, or SVM support vector machines for PdM tasks. However, complex architectures are most often used to achieve high accuracy, except for analyses related to resource-constrained industrial conditions [17]. Table 1 summarizes the research on predictive maintenance that serves as a reference point for this article and the models analyzed in it. It highlights the variety of machine learning algorithms, levels of model complexity, and application domains explored in the literature.

Table 1 summarizes a selection of publications on predictive maintenance, ranging from simple, interpretable algorithms (e.g., decision trees, logistic regression, SVM, KNN) to advanced deep learning models (LSTM, CNN, or transforms). The differences in the above studies are the level of complexity of the models and the domain of application, ranging from manufacturing to water infrastructure to aviation. The results of these studies show that classical methods often achieve high accuracy (from 85.2% to over 99%) and F1 scores above 84.8%, with low to moderate training and inference time, making them suitable for mobile and edge deployments. In contrast, more advanced approaches, i.e., deep learning, while achieving accuracy of 92.2–96.6% and F1-score of 95.9–97.1%, have higher computational resource requirements. Much of the literature

Table 1. Overview of machine learning approaches in predictive maintenance across studies and application domains

Study	Study type	Machine learning algorithms used	Model complexity level	Application domain
[18]	Empirical	Support vector machine (SVM), linear discriminant analysis (LDA), random forest (RF), decision tree (DT), k-Nearest neighbor (kNN)	Simple/interpretable	Manufacturing (bearings)
[19]	Empirical	Recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), convolutional neural network (CNN), transformer, hybrid transformer-GRU	Advanced deep learning/ hybrid	Water infrastructure (pumps)
[20]	Empirical	Logistic regression, support vector machine (SVM), random forest (RF), XGBoost, long short-term memory (LSTM)	Mixed (simple and advanced)	Industrial (ball bearings)
[12]	Empirical	Logistic regression, support vector machine (SVM), random forest (RF), XGBoost, long short-term memory (LSTM)	Mixed (simple and deep learning)	Manufacturing
[21]	Review	Deep learning architectures: Multilayer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), deep belief network (DBN), restricted Boltzmann machine (RBM), autoencoder (AE), variational autoencoder (VAE), generative adversarial network (GAN)	Advanced deep learning	Manufacturing (varied)
[22]	Empirical	Deep neural network (DNN), recurrent neural network (RNN)	Advanced deep learning	Manufacturing (machines)
[23]	Empirical	Multi-head attention (transformer), long short-term memory (LSTM)	Advanced deep learning	Manufacturing (turbofan engines)
[24]	Empirical	Compressed recurrent neural networks (RNNs)	Advanced deep learning (resource-constrained)	Manufacturing (induction motors)
[25]	Empirical	Decision tree (DT), k-Nearest neighbor (kNN), support vector machine (SVM), random forest (RF), AdaBoost, naive bayes (NB), XGBoost	Simple/interpretable	Manufacturing (smart maintenance)
[26]	Empirical	Deep learning ensemble (Neural Network), tree ensembles, support vector machine (SVM), gradient boosting machine (GBM), random forest (RF)	Mixed (simple and advanced)	Aviation (turbofan engines)

has focused on maximizing prediction accuracy without always considering the practical aspects of implementation, such as error cost, prediction and training times, or model size [27]. Despite the dynamic development of PdM methods, there remains a significant gap in the literature for evaluating models that balance predictive performance with computational power and practical approaches to implementation [28]. Especially in the context of a limited number of key input variables and implementation aspects in industrial environments with limited computing resources [29]. Much of the research relies on extensive feature sets or high-frequency sensor data, which may not be feasible in legacy systems or environments with limited data infrastructure [30].

This study fills a gap in the literature by comparing the effectiveness and efficiency of six ML algorithms (decision trees, neural networks, SVM, KNN, naive bayes, and logistic regression) for the task of machine maintenance prediction. The novelty of the study lies in the focus on three fundamental and easily measurable input variables: temperature, vibration, and machine state. In addition to standard classification metrics such as accuracy, F1-score, and AUC, the evaluation also includes practical aspects relevant to real-world industrial use, such as training time, prediction speed, and model size. This comprehensive evaluation provides insight not only into the predictive capabilities of the models, but also into their applicability in time-critical and resource-constrained environments.

RESEARCH METHODOLOGY

The objective of the modeling task was to predict whether a given machine instance requires maintenance, based on key indicators measured during its operation. The goal was to develop an accurate, efficient, and deployable classification model capable of distinguishing between normal operating conditions and those requiring preventive intervention.

All models were developed and evaluated using a consistent dataset consisting of 90,000 training observations and 10,000 test observations. Each instance included three predictors (temperature, vibration, machine condition) and a binary response variable (maintenance required) representing two classes: 0 and 1 [31]. The variable "temperature" was measured in degrees Celsius in the range of 20 °C to 120 °C. "Vibration" was measured in mm/s in the range 0.1 to 10 mm/s. The "machine condition" was coded as a discrete variable: 0 (normal condition), 1 (warning condition), 2 (emergency condition). The response variable "maintenance required" was binary (0 – no need, 1 – service intervention required).

To ensure robustness and comparability between classifiers, each model configuration was validated using a 5-fold cross-validation strategy. This approach allowed estimation of generalization performance while mitigating variance due to data partitioning. Final performance metrics were obtained on the independent test set after retraining the best performing configurations on the full training data.

The three input features used for classification represented distinct data types. Temperature and vibration were treated as continuous numerical variables and were standardized using z-score normalization to improve the stability and convergence of training. In contrast, machine status was a categorical feature encoded as integer values (0, 1, 2), reflecting different discrete operational modes of the machine. As such, this variable was excluded from standardization and directly passed to models that can natively handle categorical inputs. For classifiers requiring numerical inputs (e.g., SVM, logistic regression), the encoded form was used without further transformation.

The dataset exhibited a naturally imbalanced class distribution, with approximately 80.3% of observations labeled as class 0 (no maintenance required) and 19.7% as class 1 (maintenance required). This imbalance reflects typical industrial conditions, where failure or maintenance events are relatively rare. The same distribution was preserved in both the training and test sets. Despite this imbalance, no synthetic resampling techniques were applied. Instead, model performance was assessed using metrics sensitive to imbalance, such as the F1 score and ROC AUC, and confusion matrices were analyzed to evaluate the trade-off between false positives and false negatives.

The following classification algorithms were evaluated in this study: decision trees, neural networks, support vector machines (SVM), k-Nearest neighbors (KNN), naive bayes, and logistic regression.

The decision tree classifiers were designed to evaluate the influence of model complexity on classification accuracy. The split criterion was consistently defined using the Gini diversity index, while surrogate decision splits were disabled in all models. Three variants were examined, differing in the maximum number of splits allowed: 4, 20, and 100. These limits reflected increasing tree depth and thus increasing model capacity. The goal was to investigate the balance between underfitting and overfitting as a function of tree granularity.

Fully connected feedforward neural networks were developed in several configurations to assess the effect of architectural complexity. The number of hidden layers was varied from one to three, and the size of the hidden layers was tested with 10, 25, and 100 neurons. Each network used the ReLU activation function, and training was limited to a maximum of 1000 iterations. No regularization was applied, as the regularization parameter λ was set to zero for all models. All input features were standardized prior to training to improve the stability and convergence rate of the optimization. This range of configurations allowed the evaluation of depth and width tradeoffs in nonlinear representation learning.

Support vector machine classifiers were analyzed using different kernel functions and regularization parameters. Radial basis function (Gaussian) kernels were tested with kernel scales of 0.43, 1.7, and 6.9, while all configurations applied a constant box constraint level of 1. In addition, linear, quadratic, and cubic kernel variants were implemented with automatic kernel scaling and the same regularization settings. The classification scheme used the one-vs-one strategy for binary classification. Predictor standardization was applied throughout, with the exception of kernel expansion variants. The purpose of these experiments was to explore the ability of different kernel mappings to transform the input space for optimal separation.

The K-Nearest Neighbors classifiers were configured by varying the number of neighbors, distance metrics, and weighting schemes. The number of neighbors was set to 1, 10, or 100 to observe the effect of local versus more generalized neighborhood structures. Distance metrics included Euclidean, Minkowski with a cubic exponent, and cosine similarity, while the weight applied to neighbor contributions followed either an equal weighting scheme or a squared inverse weighting function. All KNN models were trained on standardized data to avoid scale bias. These variations facilitated a comprehensive investigation of locality-sensitive classification behavior.

Two probabilistic models based on the Naive Bayes assumption were constructed to evaluate generative approaches to classification. One model used kernel density estimation with Gaussian kernels for continuous predictors, while the other relied on the assumption of normally distributed numerical features. In both cases, no categorical variables were used, and the kernel-based variant did not include data standardization, while the Gaussian model operated under standard preprocessing conditions. These models allowed the assessment of distributional assumptions and their impact on performance in a low-dimensional feature space.

Two logistic regression models were used to represent discriminative linear classification strategies. The first, called efficient GLM logistic regression, used automatic selection for the optimization solver and regularization technique, along with a beta tolerance of 0.0001 to control convergence. The second, Binary GLM Logistic Regression, used a default configuration with no user-defined hyperparameters. Both models were used to examine the effectiveness of linear separation in binary classification under standardized data input conditions. In order to identify the best-performing classification model among the architectures tested – namely decision trees, neural networks, SVM, KNN, naive bayes, and logistic regression – a structured evaluation procedure based on multiple performance metrics was applied. These metrics were chosen to provide a comprehensive assessment of both predictive accuracy and computational efficiency, while taking into account potential class imbalance and practical deployment constraints. The primary metric used for model comparison is classification accuracy on the test set, defined as the proportion of correctly classified observations. It is calculated using the following formula:

$$Accuracy = \frac{TP=TN}{TP=TN=FP=FN} \tag{1}$$

where: TP and TN represent the counts of true positives and true negatives, while FP and FN denote false positives and false negatives, respectively.

Accuracy was evaluated on both the validation folds (via 5-fold cross-validation) and the independent test set. The test accuracy serves as the primary indicator of generalization capability.

Complementary to accuracy, the error rate quantifies the proportion of incorrect predictions and is defined as:

$$Error\ rate = 1 - Accuracy \quad (2)$$

A lower error rate reflects fewer misclassifications and provides an alternative view on model precision, particularly in high-accuracy scenarios.

To address potential limitations of accuracy in imbalanced class scenarios, the weighted F1 score was used as a secondary but critical performance metric. This score incorporates both precision and recall through the harmonic mean:

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)$$

The weighted variant adjusts the contribution of each class in proportion to its support in the data set. This is particularly useful for binary classification tasks with unequal class representation. A high F1 score indicates that the model has both low false positive and low false negative rates.

To evaluate the practical usability of each model, prediction speed (measured in observations per second), training time (in seconds), and model size (in bytes) were included as additional evaluation dimensions. High prediction speed is critical in real-time systems or large-scale deployment environments. Reduced training time and model size are beneficial for resource-constrained applications and iterative development workflows.

In addition to the scalar evaluation metrics, each classifier was evaluated using the confusion matrix, which summarizes the performance of the model by reporting the number of correctly and incorrectly classified instances for each class. For a binary classification problem with classes labeled 0 and 1, the confusion matrix takes the form shown in Figure 1, where:

- TP (true positives): the number of class 1 instances correctly predicted as class 1,
- TN (true negatives): the number of class 0 instances correctly predicted as class 0,
- FP (false positives): class 0 instances incorrectly predicted as class 1,
- FN (false negatives): class 1 instances incorrectly predicted as class 0.

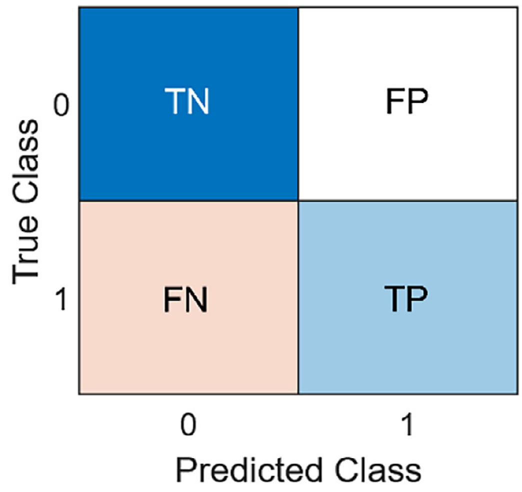

Figure 1. General model of the matrix

The selection of the six classification methods - decision trees, neural networks, support vector machines, k-Nearest neighbors, naive bayes, and logistic regression-was guided by their complementary algorithmic properties and widespread use in practical applications. Each represents a distinct family of learning paradigms: tree-based methods are interpretable and fast, neural networks are capable of capturing complex nonlinear relationships, SVMs offer strong performance in high-dimensional spaces, and KNN provides instance-based learning with minimal assumptions. Logistic regression serves as a strong linear foundation, while Naive Bayes provides probabilistic inference under independence assumptions. Together, these models provide a representative benchmark across the spectrum of complexity, interpretability, and computational cost, allowing for a thorough and balanced comparative evaluation.

RESULTS

Six machine learning models were used to predict machine maintenance needs: decision tree, naive Bayes classifier, KNN, neural network, logistic regression, and layered neural network. All models were trained on the same dataset with three input variables.

The best performing decision tree model corresponded to the coarse tree configuration, developed with a maximum of four allowed splits and using the Gini diversity index as the splitting criterion. Among neural networks, the highest accuracy was achieved by a model with a single hidden layer of 25 neurons. In the support vector machine category, the most effective model used a radial basis function (Gaussian) kernel with a kernel scale of 0.43 and a box constraint level of 1. Classification was performed using a one-vs-one multiclass coding scheme with standardized input data. The optimal k-Nearest neighbors model was configured with 10 neighbors, using the Euclidean distance metric and squared inverse distance weighting; input features were also standardized. Within the Naive Bayes group, the best results were obtained using a custom implementation based on kernel density estimation with a Gaussian kernel. Finally, the most effective logistic regression model followed the Efficient Logistic Regression setup with automatic choice of solver and regularization strategy, with automatic optimization of the regularization strength and a relative coefficient tolerance (beta tolerance) of 0.0001 to ensure convergence. The results for the models are shown in Table 2.

The highest prediction accuracy on the test set was achieved by the decision tree and neural network models, each of which exceeded 98.2% accuracy. Their results were also accompanied by the lowest error rates (1.76–1.77%) and the highest values of the F1 score measure (98.23–98.24%), indicating their high performance in maintenance needs prediction tasks.

In particular, the decision tree model offers a favorable balance between accuracy and computational efficiency. It not only achieved the highest prediction speed (more than 421,000 observations per second) and the shortest training time (5.49 s), but also the smallest model size (4702 bytes). Its speed was significantly higher than the other models, including the neural network, which, while also effective, took more than 2300 seconds to learn and had a larger model (6718 bytes).

The Neural Network model achieved very similar predictive performance (98.23% accuracy), but with a much longer training time (2372 seconds) and a larger model size (6718 bytes). Its prediction speed was about 277,000 observations per second, which also makes it practical, although less efficient than the Decision Tree model.

SVM and KNN models also provided high predictive performance, with test accuracies above 97.6% and F1 scores above 97%. Naive Bayes and logistic regression showed lower predictive performance, with test accuracies of 96.34% and 92.83%, respectively, and higher error rates. Logistic regression, although efficient in terms of learning time (~ 34 sec) and model size (11573 bytes), had the lowest F1 scores, indicating a higher tendency to misclassify minority classes - an important factor in maintenance prediction scenarios where false negatives can be costly.

The best models (decision tree, NN) achieved ≈ accuracy (~ 98.2%) in the F1 test set, confirming their balanced performance across all classes. In addition, the Tree model showed the highest computational efficiency, making it the most practical choice for predictive systems in industrial applications.

Figure 2 presents the ROC curves for all developed models, illustrating their class-separating capacity independently of the classification threshold. All models, except logistic regression, achieved AUC scores above 0.95, confirming their ability to distinguish between positive (maintenance) and negative (non-maintenance) classes. The decision tree showed an AUC = 0.9553, achieving TPR = 1 for class ‘0’ and FPR ≈ 0.089 (TPR ≈ 0.91 for class ‘1’), demonstrating excellent class separation with minimal false alarms. The neural network model (medium) achieved an AUC = 0.9543, with an operating point identical to the decision tree - TPR = 1 and FPR ≈ 0.089 for class ‘0’ and TPR ≈ 0.91 with FPR = 0 for class ‘1’, indicating high sensitivity and an excellent balance between detecting maintenance cases and avoiding false

Table 2. Evaluation metrics and runtime characteristics

Model	Decision tree	Neural network	SVM	KNN	Naive Bayes	Logistic regression
Accuracy % (validation)	98.36	98.35	98.23	97.72	96.31	92.87
Error rate % (validation)	1.64	1.65	1.77	2.28	3.69	7.13
F1 score % (validation)	98.36	98.35	98.23	97.72	96.31	92.87
Accuracy % (test)	98.24	98.23	98.19	97.67	96.34	92.83
Error rate % (test)	1.76	1.77	1.81	2.33	3.66	7.17
F1 score % (test)	98.24	98.23	98.19	97.67	96.34	92.83
Prediction speed (obs/s)	421370.61	277443.56	10818.27	28891.43	167.09	232941.02
Training time (s)	5.49	2372.12	1787.27	189.84	2288.31	34.38
Model size (bytes)	4702	6718	532087	5171525	8657033	11573

classifications. The other models also achieved an AUC > 0.95, corresponding to a strong ability to discriminate between classes, although their TPR for class '1' ranged from approximately 82% to 91%. Efficient logistic regression performed weaker (AUC = 0.9007), with an FPR ≈ 0.005, a TPR ≈ 0.66 for class '1' and a higher FPR for class '0' (~0.34), suggesting a greater trade-off between false alarms and detection of true maintenance needs.

The operating points indicated on the ROC curves (Figure 2) correspond to the default classification threshold of 0.5. These points are automatically generated by MATLAB's classification learner app based on posterior class probabilities. Manual threshold optimization was not applied.

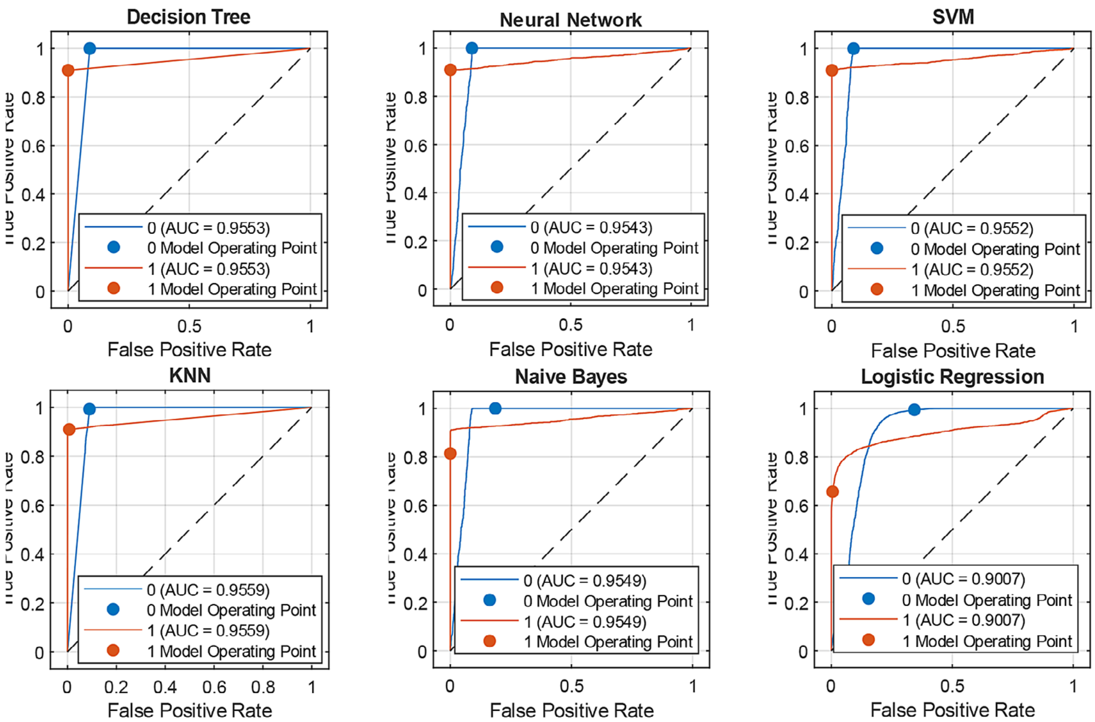

Figure 2. ROC Curves for developed models

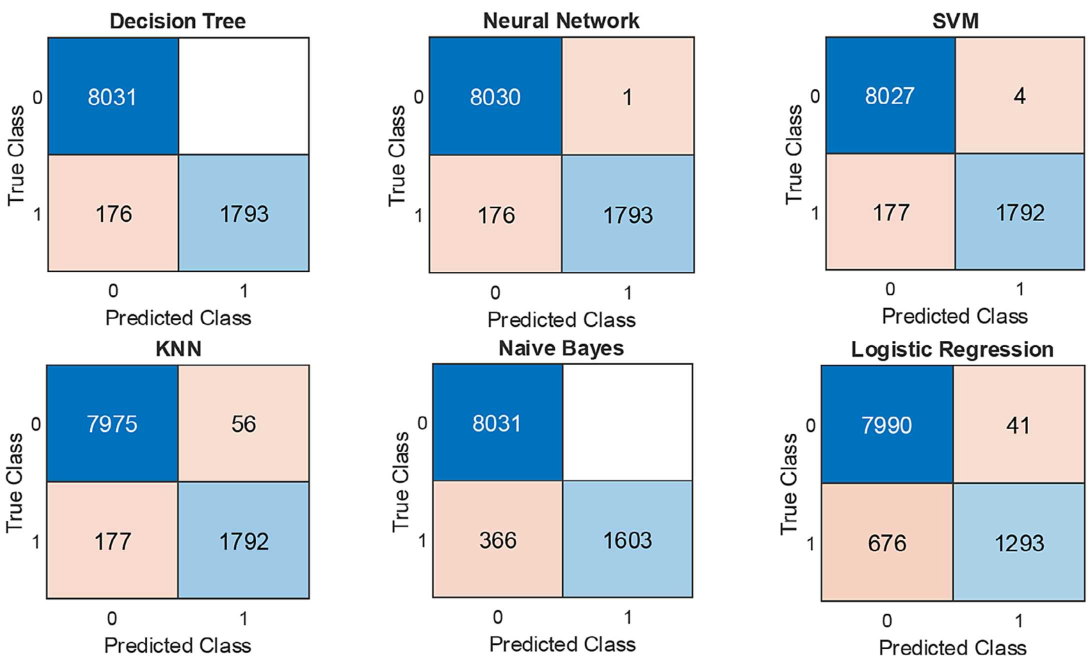

Figure 3. The confusion matrices (test dataset) for developed models

The two markers shown on each ROC curve reflect class-wise performance at that threshold (for class 0 and class 1, respectively), not different datasets. Complementing the ROC analysis, Figure 3 presents the confusion matrices obtained on the test dataset using the default threshold (0.5). These matrices provide insight into the actual classification behavior of each model.

Both decision tree and neural network showed almost identical predictive behavior: each model made 176 false negative (FN) classification errors, i.e. cases where the system did not recognize a real maintenance need (class 1 classified as 0). The number of false positive (FP) errors, i.e. unjustified predictions of maintenance needs, was close to zero – 0 for decision tree and 1 for NN – indicating very high accuracy relative to the negative class.

The SVM model recorded 177 cases of FN and 4 cases of FP, indicating a slightly higher tendency to misclassify both cases of actual maintenance need and its unjustified prediction. The KNN model showed a total of 233 misclassifications – 177 FN and 56 FP. Of particular concern is the number of false alarms, which can lead to unnecessary maintenance and operational costs.

In the case of Naive Bayes, although no FP cases (i.e., predicting maintenance when it was not needed) were recorded, the model made as many as 366 FN errors, meaning that a significant number of actual maintenance needs were missed – making the model particularly risky in applications requiring high sensitivity. The logistic regression model had the highest number of errors (717) and the lowest accuracy for the test set at 92.83%. The high FN (676) indicates a significant risk of missing maintenance cases. All models showed an advantage in negative classification (‘0’ classes), but differed in their ability to capture positive cases. Decision trees and neural networks achieved the lowest FP and FN numbers, resulting in the highest overall performance.

The decision tree model showed the best overall performance for predictive maintenance. It achieved the highest accuracy (98.24%), the lowest error rate (1.76%), and the highest F1 score (98.24%), while maintaining an excellent AUC of 0.9553. Its confusion matrix showed no false positives and minimal false negatives, confirming high precision and recall. What sets it apart is its exceptional efficiency: the shortest training time (5.49 s), the smallest model size (4702 bytes), and the fastest prediction speed (over 421,000 obs/s), making it ideal for real-time industrial applications. Although the neural network achieved similar accuracy, it required much longer training time and slower inference speed. Other models performed well but had trade-offs in size, speed, or sensitivity. Overall, the decision tree stands out as the most accurate, efficient, and practical solution for maintenance prediction.

DISCUSSION

The objective of this study was to identify the most effective machine learning algorithm for the task of predicting machine maintenance needs based on three input variables. A comprehensive comparison of six classifiers was performed: decision tree, neural network, SVM, KNN, Naive Bayes classifier, and logistic regression. Under the conditions of a consistent data set and uniform cross-validation, the decision tree model proved to be the most balanced and effective solution.

Decision Tree achieved the highest classification accuracy (98.24%), the lowest error rate (1.76%) and the highest F1 score (98.24%). Confusion matrix analysis showed that the model generated no false alarms (FP = 0) and made only 176 FN errors, confirming its high sensitivity in detecting actual maintenance needs. In addition, the model had the highest AUC (0.9553) and an ideal TPR = 1 for the “0” class, demonstrating its excellent separation ability. At the same time, the computational efficiency of the decision tree model significantly outperformed the other methods, with a training time of only 5.49 s, a model size of 4702 bytes, and a prediction speed of more than 421 000 observations per second. This makes the model particularly attractive for industrial applications where not only relevance but also response time and resource efficiency are important.

The neural network achieved comparable classification performance (98.23% accuracy, F1), but required much longer training (> 2.300 s) and had a larger model size, which limits its operational efficiency. SVM and KNN models also achieved high AUCs (> 0.95), but at the expense of low prediction speed (SVM) or large model size and higher false alarms (KNN). In contrast, Naive Bayes and logistic regression showed lower performance in terms of accuracy, F1 and number of errors. In particular, logistic regression proved to be the least suitable for detecting maintenance cases, with a high FN (676) and the lowest AUC (0.9007).

The results of this study are part of current research trends to optimize predictive models for industrial applications [25–27]. The demonstrated superiority of simple interpretable models over complex deep learning architectures is confirmed by work on edge device implementations, where models such as decision trees or SVMs achieve over 97% accuracy with minimal resource consumption [35, 36]. Similarly, studies integrating predictive maintenance with ERP systems (e.g., SAP, QAD) highlight the key role of model inference speed and compactness, especially in the context of real-time processing of IoT sensor data [34, 37]. In contrast to work focusing on hybrid deep learning models, this study confirms that even with a limited number of input variables, simple algorithms can compete with neural networks in terms of efficiency, which is particularly important in low-latency environments (e.g., manufacturing systems) [38, 39]. These results are consistent with recommendations from literature reviews that point to the increasing importance of computational efficiency in PdM, even at the expense of a marginal decrease in accuracy [40].

The study provides both practical and theoretical conclusions regarding the effectiveness of selected machine learning algorithms in predictive maintenance tasks. The results clearly indicate that models with low computational complexity - such as decision trees - can provide performance comparable to more complex models, while significantly reducing training and prediction costs [41]. This is an important observation from the point of view of designing machine monitoring systems, especially in industrial environments where computational resources are limited or real-time operation is required.

In many industrial classification tasks, such as predictive maintenance, datasets are inherently imbalanced due to the rarity of failure events. While this study evaluated models without resampling, future improvements could consider techniques like SMOTE or cost-sensitive learning. These methods, as shown by Awtoniuk et al., can significantly enhance minority class detection even under severe imbalance conditions [42].

From a theoretical point of view, the study confirms that with well-chosen input features and properly performed parameter selection, classical machine learning methods are still strong competitors for more complex solutions, especially in the context of small or medium complex datasets. This points to the need for further research on the trade-off between complexity and model performance, especially in industrial applications where reliability and speed are critical.

CONCLUSIONS

The aim of this paper is to comprehensively evaluate the effectiveness and usefulness of selected machine learning algorithms in the task of predicting the maintenance needs of industrial machinery. Six models representing different classification approaches were analyzed: decision tree, neural network, SVM, KNN classifier, naive Bayes classifier, and logistic regression. The models were trained and tested on a common dataset containing only three simple predictors: temperature, vibration and machine state, allowing their performance to be evaluated under conditions of limited information.

The evaluation was not only based on classical performance measures such as prediction accuracy, F1 measure and area under the ROC curve (AUC), but also took into account practical aspects such as training time, prediction speed and model size. In addition, the analysis of the confusion matrix allowed the evaluation of false positive (FP) and missed maintenance (FN) errors, which is crucial in the context of real industrial applications.

The study showed that decision tree and neural networks (two-layer and medium) achieve the best performance in predicting maintenance needs, with an AUC > 0.95, an F1-score of 97–99% and the lowest misclassification costs. Decision Tree is also characterized by minimal computational requirements (training time of 5.5 s, more than 420,000 objects/s prediction, model size < 5 KB), making it ideal for real-time systems.

The use of decision trees can significantly reduce operational costs and unplanned downtime, while neural networks can provide slightly higher accuracy in highly variable data environments, despite their higher resource requirements.

Despite the high relevance of the results obtained, this study has some limitations. First, the analyses were performed on a synthetic data set with a fixed structure and a limited number of input variables (temperature, vibration, machine state), which could have simplified the classification problem. Second, all models were tested in an offline environment, without taking into account dynamic changes of the data over time or model degradation under real conditions (concept drift). Finally, the selection of architectures and learning parameters was done manually, which may limit the optimization potential of more complex models such as neural networks or SVMs.

In future studies, it would be worthwhile to extend the analysis to real-world data from industrial monitoring systems, including temporal and sequential variables such as equipment operating history or service intervention data. The use of ensemble techniques (e.g. random forest, gradient boosting) could further improve the classification performance, especially in the detection of minority classes. It is also recommended to automate the selection of hyperparameters (e.g. through Bayesian optimization or grid search), which could increase the potential of higher complexity models. Importantly, future work may explore rebalancing strategies to further improve the detection of minority class instances. This includes techniques such as SMOTE (synthetic minority oversampling technique), adaptive resampling, cost-sensitive learning, and hybrid ensemble methods like RUSBoost. These approaches have proven effective in recent studies on industrial fault prediction, where early detection of rare failure events is critical to minimizing unplanned downtime and ensuring system reliability. An important direction of development will also be the evaluation of models in an online setting, taking into account the real-time data variability, to better reflect the operational challenges of implementing predictive maintenance systems in real production facilities.

REFERENCES

1. Kulisz M., Kujawska J., Cioch M., and Cel W. Modeling energy access challenges in europe: a neural network approach to predicting household heating inadequacy using macro-energy indicators, Energies, Dec. 2024, 17(23), 6104, https://doi.org/10.3390/en17236104
2. Przysucha B., Kulisz M., Kujawska J., Cioch M., Gawryluk A., and Garbacz R. Modeling ecological risk in bottom sediments using predictive data analytics: Implications for energy systems, Energies, May 2025, 18(9), 2329, https://doi.org/10.3390/en18092329
3. Tarik M., Mniai A., and Jebari K. Hybrid feature selection and support vector machine framework for predicting maintenance failures, acs, Jun. 2023, 19(2), 112–124, https://doi.org/10.35784/acs-2023-18.
4. Pizoń J., Cioch M., Kański Ł., and Sánchez García E. Cobots implementation in the era of industry 5.0 using modern business and management solutions, Adv. Sci. Technol. Res. J., Dec. 2022, 16(6), 166–178, https://doi.org/10.12913/22998624/156222
5. Cioch M. Forecasting the outsourcing use in small and medium-sized enterprises: A multiple regression model, ERSJ, XXVII, Special Issue B, Jul. 2024, 456–472, https://doi.org/10.35808/ersj/3500
6. Cioch M. and Kulisz M. The use of fuzzy logic in the decision-making process regarding the use of outsourcing in companies, JoMS, Dec. 2023, 54(5), 66–82, https://doi.org/10.13166/jms/176173
7. Cioch M., Kulisz M., and Kański Ł. Implementing AI collaborative robots in manufacturing – modeling enterprise challenges in industry 5.0 with fuzzy logic, Adv. Sci. Technol. Res. J., Nov. 2024, 18(7), 229–238, https://doi.org/10.12913/22998624/192833
8. Józwik J. et al., Modeling the optimal measurement time with a probe on the machine tool using machine learning methods', Appl. Comput. Sci., Jun. 2024, 20(2), 43–59, https://doi.org/10.35784/acs-2024-15
9. Sukop M., Grytsiv M., Jánoš R., and Semjon J. Simple ultrasonic-based localization system for mobile robots, Applied Sciences, Apr. 2024, 14(9), 3625, https://doi.org/10.3390/app14093625
10. Semjon J., Hajduk M., Jánoš R., and Vagaš M. Modular welding fixtures for robotic cells, AMM, Feb. 2013, 309, 80–87, https://doi.org/10.4028/www.scientific.net/amm.309.80
11. Kane A. P., Kore A. S., Khandale A. N., Nigade S. S., and Joshi P. P. Predictive maintenance using machine learning, 2022, arXiv. https://doi.org/10.48550/ARXIV.2205.09402
12. Wadibhasme R. N., Naresh M., Vikram G., A. S. V V, S. P, and Jermina F. Utilizing machine learning techniques for enhanced predictive maintenance in the manufacturing sector, in 2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS), IEEE, Aug. 2024, 1–6. https://doi.org/10.1109/icnews60873.2024.10730983
13. Deptula A., Natorska M., Prażnowski K., Mamala J., and Urbanowicz K. Application of a decision classifier tree to evaluate energy consumption of an electric vehicle under real traffic conditions, Adv. Sci. Technol. Res. J., Apr. 2025, 19(4), 91–108, https://doi.org/10.12913/22998624/199799
14. Mohd Kamal M. H., Isham M. F. B., Raheimi A., Mohd Saufi M. S. R., and Abdul Saad W. A. Bearing fault diagnosis based on cross-machine statistical features generalization and improved

extreme learning machine, Adv. Sci. Technol. Res. J., May 2025, 19(5), 407–421, https://doi.org/10.12913/22998624/202742

15. Mahida A. Predictive incident management using machine learning, IJSR, Jun. 2022, 11(6), 1977–1980, https://doi.org/10.21275/SR24401231847
16. Mennilli R., Mazza L., and Mura A. Integrating machine learning for predictive maintenance on resource-constrained PLCs: A feasibility study, Sensors, Jan. 2025, 25(2), 537, https://doi.org/10.3390/s25020537
17. Carvalho T. P., Soares F. A. A. M. N., Vita R., Francisco R. D. P., Basto J. P., and Alcalá S. G. S. A systematic literature review of machine learning methods applied to predictive maintenance, Computers & Industrial Engineering, Nov. 2019, 137, 106024, https://doi.org/10.1016/j.cie.2019.106024
18. Cakir M., Guvenc M. A., and Mistikoglu S. The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system, Computers & Industrial Engineering, Jan. 2021, 151, 106948, https://doi.org/10.1016/j.cie.2020.106948
19. Dintén R. and Zorrilla M. Design, building and deployment of smart applications for anomaly detection and failure prediction in industrial use cases, Information, Sep. 2024, 15 (9), 557, https://doi.org/10.3390/info15090557
20. Farooq U., Ademola M., and Shaalan A. Comparative analysis of machine learning models for predictive maintenance of ball bearing systems, Electronics, Jan. 2024, 13(2), 438, https://doi.org/10.3390/electronics13020438
21. Serradilla O., Zugasti E., Rodriguez J., and Zurutuza U. Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects, Appl Intell, Jan. 2022, 52(10), 10934–10964, https://doi.org/10.1007/s10489-021-03004-y
22. Zonta T., da Costa C. A., Zeiser F. A., de Oliveira Ramos G., Kunst R., and da Rosa Righi R. A predictive maintenance model for optimizing production schedule using deep neural networks, Journal of Manufacturing Systems, Jan. 2022, 62, 450–462, https://doi.org/10.1016/j.jmsy.2021.12.013
23. De Luca R., Ferraro A., Galli A., Gallo M., Moscato V., and Sperlì G. A deep attention based approach for predictive maintenance applications in IoT scenarios, JMTM, Feb. 2023, 34(4), 535–556, https://doi.org/10.1108/jmtm-02-2022-0093
24. Markiewicz M., Wielgosz M., Bochenski M., Tabaczynski W., Konieczny T., and Kowalczyk L. Predictive maintenance of induction motors using ultra-low power wireless sensors and compressed recurrent neural networks, IEEE Access, 2019, 7, 178891–178902, https://doi.org/.1109/access.2019.2953019
25. Bodo R., Bertocco M., and Bianchi A. Fault classification driven by maintenance management for smart maintenance applications, in 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, IEEE, Jun. 2020, 27–32. https://doi.org/10.1109/metroind4.0iot48571.2020.9138294
26. Matuszczak M., Żbikowski M., and Teodorczyk A. Predictive modelling of turbofan engine components condition using machine and deep learning methods, Eksploatacja i Niezawodność – Maintenance and Reliability, Jun. 2021, 23(2), 359–370, https://doi.org/10.17531/ein.2021.2.16
27. Moulla D. K., Mnkandla E., Aboubakar M., Adamou A. A. A., and Abran A. 'PdM-FSA: predictive maintenance framework with fault severity awareness in Industry 4.0 using machine learning', IJECE, Dec. 2024, 14(6), 7211, https://doi.org/10.11591/ijece.v14i6.pp7211-7223
28. Zhu T., Ran Y., Zhou X., and Wen Y. A survey of predictive maintenance: systems, purposes and approaches, 2019, arXiv. https://doi.org/10.48550/ARXIV.1912.07383
29. Filios G., Katsidimas I., Nikoletseas S., Panagiotou S. H., and Raptis T. P. Agnostic learning for packing machine stoppage prediction in smart factories, ITU J-FET, Dec. 2022, 3(3), 793–807, https://doi.org/10.52953/LEDZ3942
30. Cummins L. et al. Explainable predictive maintenance: a survey of current methods, challenges and opportunities, 2024, arXiv. https://doi.org/10.48550/ARXIV.2401.07871
31. Ilyas M. 'Predictive_Maintenance_Using_ANN'. kaggle. [Online]. Available: https://www.kaggle.com/code/ilyass323/predictive-maintenance-using-ann/notebook
32. Ronkiewicz T., Aleksiejuk-Gawron J., Awtoniuk M., and Kurek J. Neural modelling of solar radiation variability, J. Phys.: Conf. Ser., Jan. 2021, 1736(1), 012015, https://doi.org/10.1088/1742-6596/1736/1/012015
33. Kłosowski G., Hoła A., Rymarczyk T., Skowron Ł., Wołowiec T., and Kowalski M. The concept of using LSTM to detect moisture in brick walls by means of electrical impedance tomography, Energies, Nov. 2021, 14(22), 7617, https://doi.org/10.3390/en14227617
34. Pizoń J., Wójcik Ł., Gola A., Kański Ł., and Nielsen I. Autonomous mobile robots in automotive remanufacturing: a case study for intra-logistics support, Adv. Sci. Technol. Res. J., Feb. 2024, 18(1), 213–230, https://doi.org/10.12913/22998624/177398
35. Kulisz M., Kujawska J., Cioch M., Cel W., and Pizoń J. Comparative analysis of machine learning methods for predicting energy recovery from waste, Applied Sciences, Apr. 2024, 14(7), 2997, https://doi.org/10.3390/app14072997
36. Falih B. S., Gierz Ł., and Al-Sammarraie M. A. J.

Fruit classification by assessing slice hardness based on RGB imaging. Case study: apple slices, J. Appl. Math. Comput. Mech., Sep. 2024, 23(3), 7–18, https://doi.org/10.17512/jamcm.2024.3.01

37. Brzozowska J., Pizoń J., Baytikenova G., Gola A., Zakimova A., and Piotrowska K. Data engineering in crisp-dm process production data – case study, Appl. Comput. Sci., Sep. 2023, 19(3), 83–95, https://doi.org/10.35784/acs-2023-26
38. Kulisz M., Kłosowski G., Rymarczyk T., Słoniec J., Gauda K., and Cwynar W. Optimizing the neural network loss function in electrical tomography to increase energy efficiency in industrial reactors, Energies, Jan. 2024, 17(3), 681, https://doi.org/10.3390/en17030681
39. Antosz K., Jasiulewicz - Kaczmarek M., Machado J., and Relich M. Application of principle component analysis and logistic regression to support six sigma implementation in maintenance, Eksploatacja i Niezawodność – Maintenance and Reliability, Nov. 2023, 25(4), https://doi.org/10.17531/ein/174603
40. Singgih M. L., Zakiyyah F. F., and Andrew. Machine Learning for Predictive Maintenance: A Literature Review, in 2024 Seventh International Conference on Vocational Education and Electrical Engineering (ICVEE), Malang, Indonesia: IEEE, Oct. 2024, 250–256. https://doi.org/10.1109/ICVEE63912.2024.10823713
41. Antosz K., Kulisz M., and Husar J. Evaluation and comparison of selected machine learning methods for improving maintenance processes, IFAC-PapersOnLine, 2024, 58(8), 85–90, https://doi.org/10.1016/j.ifacol.2024.08.055
42. Awtoniuk M., Majerek D., Myziak A., and Gajda C. Industrial application of deep neural network for aluminum casting defect detection in case of unbalanced dataset, Adv. Sci. Technol. Res. J., Nov. 2022, 16(5), 120–128, https://doi.org/10.12913/22998624/154963