Journal of Advances in Manufacturing Engineering
Web page info: https://jame.yildiz.edu.tr
DOI: 10.14744/ytu.jame.2025.00008

Original Article

Comparative study of machine learning and ensemble learning approach on tool wear classification

Muhammet Ali AYKANAT*[1], Rifat KURBAN[2]

[1]*Department of Electrical and Computer Engineering, Abdullah Gül University, Kayseri, Türkiye*
[2]*Department of Computer Engineering, Abdullah Gül University, Kayseri, Türkiye*

ARTICLE INFO

Article history
Received: 15 January 2025
Revised: 02 May 2025
Accepted: 29 September 2025

Key words:
Classification, ensemble learning, machine learning, machine tools, tool wear prediction.

ABSTRACT

Tool wear is a critical challenge in machining operations, significantly affecting production quality, operational efficiency, and maintenance costs. Traditional approaches, such as sensor-based monitoring and material coatings, have limitations in accurately and proactively predicting tool wear in dynamic manufacturing environments. To address these challenges, this study explores the application of machine learning and ensemble learning methods to improve the reliability and accuracy of tool wear classification. We implemented five different algorithms: K-Nearest Neighbors (KNN), Decision Trees, Random Forests, LightGBM, and XGBoost, to predict the tool condition as "worn" or "unworn." Despite high individual model performances, each exhibited certain weaknesses, motivating the development of an ensemble learning approach. A soft voting classifier, combining KNN, Random Forest, and LightGBM, was proposed to overcome these shortcomings by leveraging the strengths of multiple models. Experimental results demonstrated that the ensemble method achieved superior performance, with an accuracy of 0.9968 on the unseen test dataset. This research highlights the potential of ensemble learning to provide robust, accurate, and generalizable solutions for tool wear prediction, contributing to smarter, more proactive maintenance strategies in manufacturing environments.

Cite this article as: Aykanat, M. A., & Kurban, R. (2025). Comparative study of machine learning and ensemble learning approach on tool wear classification. *J Adv Manuf Eng*, 6(2), 00–00.

INTRODUCTION

Tool wear is a critical issue in machining operations as it directly impacts the workpiece's quality and the machining process's efficiency [1]. The gradual loss of material from the cutting tool due to friction and other factors not only affects the tool itself but also leads to changes in the machined surface and the overall performance of the machine tool Understanding and effectively managing tool wear is essential to maintaining production quality, reducing production time, and minimizing economic losses associated with tool replacement and poor workpiece quality [2].

Researchers have explored various traditional methods and technologies to address tool wear problems without resorting to machine learning or artificial intelligence. One approach involves using sensor fusion strategies to monitor cutting tool wear [2]. By integrating data from different sensors that capture information on tool conditions during machining processes, operators can make informed decisions regarding tool replacement and maintenance to ensure consistent workpiece quality and production efficiency. Additionally, the application of Ti/AlTiN multilayer coatings on cutting tools has been investigated to mitigate the crater wear process and improve the tribological properties of

***Corresponding author.**
*E-mail address: muhammetali.aykanat@agu.edu.tr

the tools [3]. These coating technologies offer a preventive measure against wear, enhancing the durability and performance of cutting tools in machining operations. On the other hand, leveraging machine learning techniques for tool wear classification has shown promising results in enhancing the accuracy and efficiency of wear monitoring systems. Studies have demonstrated using support vector machine (SVM) algorithms coupled with time and frequency domain analysis to correlate sound signals generated during cutting processes with tool wear conditions [4]. Training machine learning models on these acoustic signatures makes it possible to classify tool wear states in real time, enabling proactive maintenance and replacement strategies to be implemented.

Furthermore, the integration of machine learning classification models, such as convolutional neural networks (CNNs), has been explored for online tool wear classification during machining processes [5]. By utilizing real-time cutting force measurements and CNN approaches, researchers have achieved significant accuracy rates in classifying tool wear states, enabling timely identification and mitigation strategies to be deployed [5]. Additionally, the use of pre-trained CNNs for vision-based tool wear classification has been investigated, highlighting the importance of timely identification and classification of wear conditions to guide tool replacement decisions and minimize wear-related issues [6].

In conclusion, the problem of tool wear in machine tools is a multifaceted issue that requires a comprehensive approach for effective management. While traditional methods like sensor fusion and coating technologies offer preventive measures against wear, the use of machine learning and artificial intelligence techniques provide advanced capabilities for real-time wear monitoring and classification. By combining these approaches, manufacturers can optimize tool usage, enhance production efficiency, and ensure consistent quality in machining operations.

In this study, various machine learning algorithms are implemented to address the tool wear problem. By leveraging the capabilities of machine learning, it becomes possible to predict tool wear with higher accuracy and reliability compared to traditional methods. The algorithms used in this study include K-Nearest Neighbors (KNN), Decision Tree, Random Forest, LightGBM, and XGBoost, each known for their unique strengths in handling different aspects of data. These models are compared in terms of their predictive accuracy to identify the most effective approach for tool wear prediction. Additionally, ensemble learning techniques are employed to combine the strengths of multiple models, aiming to achieve more robust and reliable results. Ensemble learning, through methods like voting classifiers, enhances the overall performance by mitigating the weaknesses of individual models, thus providing a more comprehensive solution to the tool wear problem.

MATERIALS AND METHODS

Dataset

The dataset, originating from the University of Michigan's System-level Manufacturing and Automation Research Testbed (SMART) published in April 2018, 18 different machining experiments performed on wax blocks (2" x 2" x 1.5") with S shape using a CNC milling machine [7]. The general data from each of the 18 distinct experiments encompass the experiment number, the material used (wax), the feed rate, and the clamping pressure. Each experiment's outputs include the condition of the tool (unworn or worn) and whether the tool passed a visual inspection. Time series data were collected from the 18 experiments at a sampling rate of 100 ms and are individually documented in files named experiment_01.csv to experiment_18.csv. Each file contains measurements from the CNC machine's four motors (X, Y, Z axes, and spindle). These experiments varied tool conditions, feed rates, and clamping pressures to investigate their effects on machining performance. The aggregated dataset comprised 25,286 observations and 52 features, of which 12 were categorical, and 40 were numerical.

Proposed Method

The proposed method, given in Figure 1, leverages a machine learning and ensemble learning approach to solve the given problem. This methodology comprises three main steps: data preprocessing, model implementation, and ensemble approach.

Data preprocessing is a crucial step that involves handling outliers and missing values, encoding categorical variables, standardizing the features, and performing stratified data splitting. Outlier handling ensures that extreme values do not skew the model's performance while addressing missing values, which prevents the introduction of bias. Encoding categorical variables transforms them into a numerical format suitable for machine learning algorithms. Standardization ensures that the features have a mean of zero and a standard deviation of one, essential for the proper convergence of many machine learning algorithms. Stratified splitting ensures that the train and test sets have similar distributions of the target variable, maintaining the representativeness of the data.

Five different machine learning models are implemented to identify the best solution: K-Nearest Neighbors (KNN) [8], Decision Tree [9], Random Forest [10], LightGBM [11], and XGBoost [12]. Each base model undergoes hyperparameter optimization and is evaluated using 5-fold cross-validation on the training set to ensure robust performance and prevent overfitting. KNN is known for its simplicity and effectiveness in classification tasks [13]. Decision Trees provide interpretability by creating a tree-like structure of decisions [14]. Random Forest, an ensemble of Decision Trees, improves performance through averaging, which reduces variance and prevents overfitting [15]. LightGBM and XGBoost are gradient-boosting frameworks that build models sequentially, with each new model correcting errors made by the previous ones [11, 12]. These methods are compelling for large datasets and have been shown to achieve high predictive accuracy [16, 17].

The ensemble approach employs a voting classifier, evaluated on the test set. The voting classifier combines KNN, Random Forest, and LightGBM as voters. Ensemble methods are known to improve predictive performance by combining the strengths of multiple models [18]. This approach reduces the

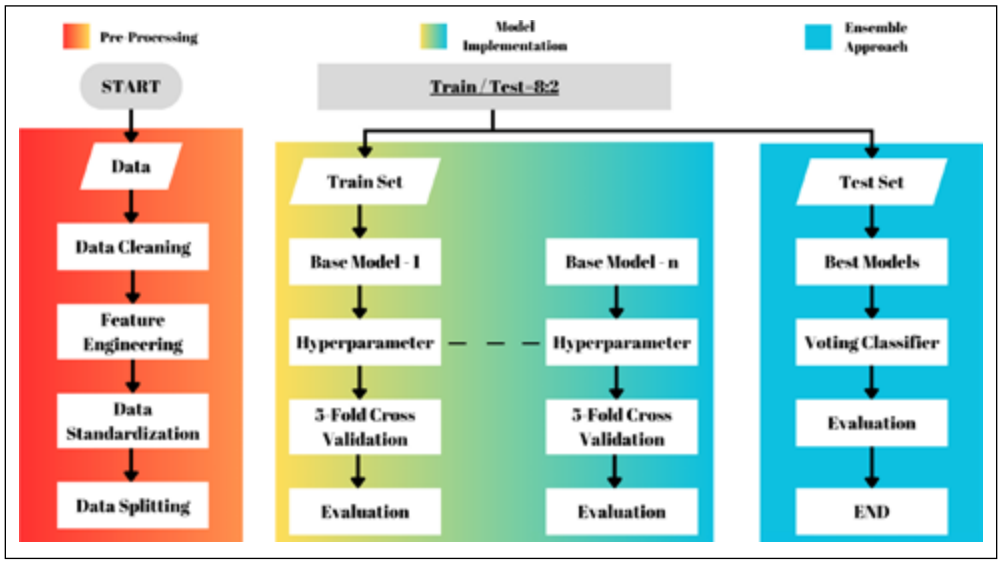

Figure 1. An architecture of proposed method.

likelihood of overfitting and increases robustness and generalization [19]. By aggregating the predictions of diverse models, the ensemble method can achieve higher accuracy and better generalization compared to individual models [20, 21].

RESULTS AND DISCUSSION

Data Preprocessing

The dataset comprised 18 experimental CSV files and one training file containing tool status labels categorized as "worn" or "unworn". The initial step involved merging the 18 experimental files into a single dataset. This merged dataset included features from the experimental files along with additional columns for exp_no, feedrate, clamp_pressure, and tool_condition extracted from the training file.

The aggregated dataset consisted of 25,286 observations and 52 features. Among these features, 12 were categorical, and 40 were numerical.

Outliers were detected in 27 features and addressed using the Interquartile Range (IQR) method to ensure a more robust dataset for analysis.

To prepare the dataset for machine learning algorithms, we meticulously applied label encoding to the tool_condition feature. This process converted the categorical labels "worn" and "unworn" into numerical values, ensuring the accuracy of the data. One-hot encoding was then applied to the other categorical features to avoid any ordinal relationships being implied by the model.

After implementing the encoding, the shape of the dataset was transformed to (25,286, 61), reflecting the addition of new columns from the one-hot encoding process. To standardize the dataset, Min-Max scaling (1) was applied to all features, bringing them into the range [0, 1]. The exp_no feature was subsequently dropped to prevent potential issues with high correlation.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

These preprocessing steps resulted in a clean, normalized, and well-structured dataset ready for subsequent machine learning model development and analysis. After pre-processing, the whole dataset was shuffled and split into different CSV files named train and test to ensure the model did not see the data.

Base Model

Before the training phase, the dataset was stratified and split into training and testing sets with an 80-20 ratio, ensuring that both sets' class distribution of the tool_condition labels was preserved. A fixed random state was used to ensure the reproducibility of the results.

Five different machine learning models, KNN, DT, RF, LightGBM, and XGBoost, respectively, were implemented to predict the tool condition of the dataset.

To assess the models' performance and their ability to generalize to unseen data, a 5-fold cross-validation was conducted on the training set. This strategy ensured that each model was trained and validated on different portions of the data, providing a solid evaluation of the model's effectiveness. The results of this evaluation are presented in Table 1.

Hyperparameter Optimization

The same split data and model were used to implement hyperparameter optimization. A 5-fold cross-validation was performed during the training phase to evaluate the models. Hyperparameter optimization was then conducted using the following ranges in Table 2. The hyperparameters' ranges were found and decided by trial and error.

The performance of each model was evaluated based on accuracy, F1-score, and ROC_AUC on the test set. The results of the best models after hyperparameter optimization with the train set are summarized in Table 3.

Model Evaluation

Accuracy measures how correct a model's predictions are overall. It is calculated as the ratio of correctly predicted instances to the total number of instances in the dataset. The formula for accuracy is:

Table 1. Base model train phase results

Model	Accuracy	F1_Score	ROC_AUC
KNN	0.8901	0.8952	0.9539
Decision Tree	0.986	0.9866	0.986
Random Forest	0.9923	0.9926	0.9998
LightGBM	0.9941	0.9944	0.9994
XGBoost	0.9942	0.9945	0.9998

Table 2. Models and their hyperparameter ranges

Model	Hyperparameter	Range
KNN	Number of neighbors	2 to 50
Decision Tree	Maximum depth	1 to 20
	Minimum sample split	2 to 30
Random Forest	Maximum depth	8 to 15
	Minimum sample split	15 to 20
	Number of estimators	200, 300
LightGBM	Learning Rate	0.01 to 0.1
	Number of estimators	300, 500
XGBoost	Learning rate	0.01 to 0.1
	Maximum depth	5 to 8
	Number of estimators	100, 200

Table 3. Models and their hyperparameter results

Model	Accuracy	F1_Score	ROC_AUC	Best parameters
KNN	0.9081	0.9124	0.9576	{'n_neighbors': 3}
Decision Tree	0.9859	0.9865	0.9901	{'max_depth': 19, 'min_samples_split': 7}
Random Forest	0.991	0.9914	0.9997	{'max_depth': None, 'min_samples_split': 15, 'n_estimators': 300}
LightGBM	0.9957	0.9959	0.9996	{'learning_rate': 0.1, 'n_estimators': 300}
XGBoost	0.9946	0.9949	0.9998	{'learning_rate': 0.1, 'max_depth': 8, 'n_estimators': 200}

$$Acc = \frac{TP+\mathrm{TN}}{All}$$

Accuracy is a valuable metric when the classes are balanced, as it provides a straightforward measure of how often the model is correct.

The F1-Score, which is the harmonic mean of precision and recall, serves as a metric that balances false positives and false negatives. It is particularly beneficial for imbalanced datasets because it takes into account both precision (the correctness of positive predictions) and recall (the capability to identify all positive cases). The formula for the F1-score is:

$$F1_{Score} = 2\,x\,\frac{\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}$$

where

$$Precision = \frac{TP}{TP+\mathrm{FP}}$$

$$Recall = \frac{TP}{TP+\mathrm{FN}}$$

The receiver Operating Characteristic (ROC) and the Area Under the Curve (AUC) are fundamental concepts in evaluating the performance of binary classification models in machine learning and statistics.

The ROC curve is a graphical representation that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is a plot of the True Positive Rate (TPR) (also called sensitivity) against the

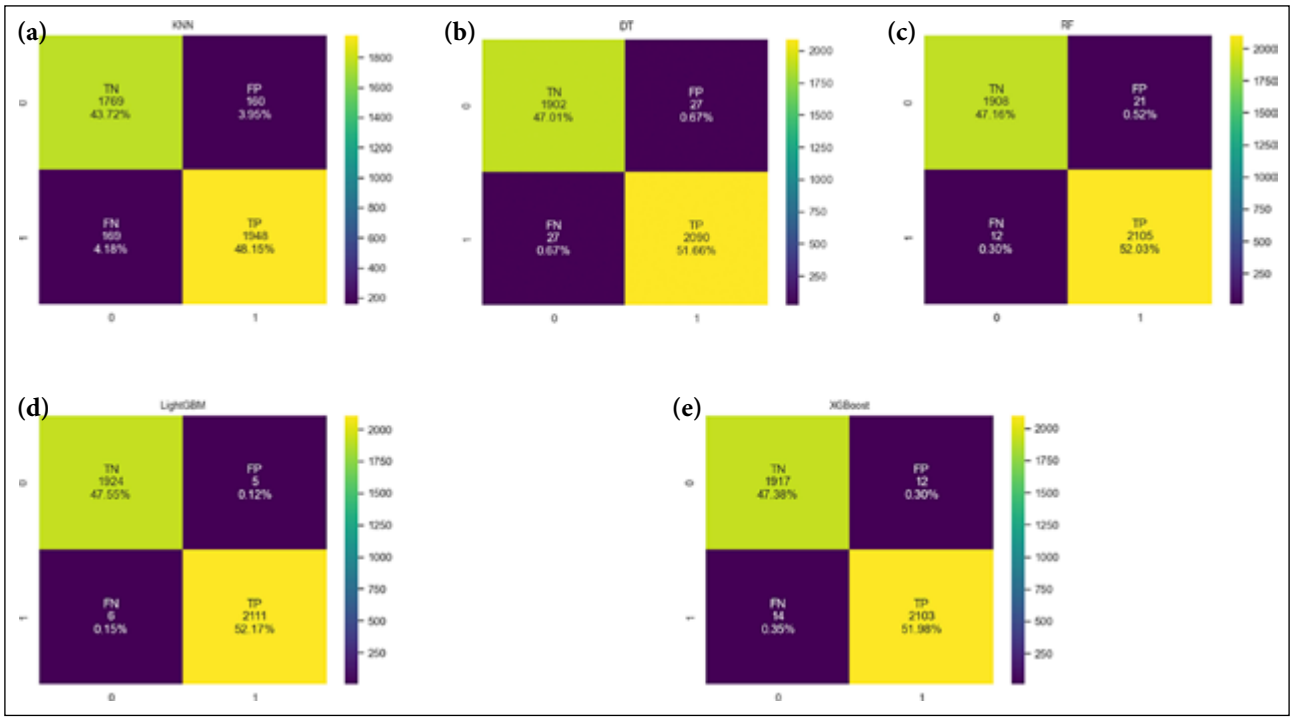

Figure 2. Confusion matrix of 5 models on train-validation sets. KNN **(a)**, DT **(b)**, RF **(c)**, LightGBM **(d)**, XGBoost **(e)**.

False Positive Rate (FPR) (also called false-alarm-rate). So, it is a commonly used summary for assessing the tradeoff between sensitivity and specificity [22]. In mathematical terms, the ROC described as:

$$TPR = \frac{TP}{TP+\mathrm{FN}}$$

$$FPR = \frac{FP}{FP+\mathrm{TN}}$$

The AUC is that summarizes how well a classification model separates positive and negative classes. It is derived from the ROC curve, which plots the true positive rate against the false positive rate at different thresholds. That's why it called ROC_AUC. When the ROC_AUC is 1.0, it indicates that the model perfectly separates the classes, making accurate predictions without any mistakes. An ROC_AUC of 0.5 means the model has no discriminative power and performs as if it were guessing randomly, with no better performance than chance. When the ROC_AUC is less than 0.5, the model performs worse than random guessing, making more incorrect predictions than correct ones, and thus is considered worse than random. The higher the AUC, the better the model is at correctly distinguishing between the classes. Mathematical formula for ROC_AUC is:

$$ROC\_AUC = \int TPR \times d(FPR)$$

where TPR is true positive rate, FPR is false positive rate

Accuracy was selected as the primary evaluation metric for this study because the dataset is close to balanced, with 13,308 instances labelled as "worn" (52.63%) and 11,978 instances labelled as "unworn" (47.37%). In a balanced dataset, accuracy provides a clear and straightforward measure of model performance, as it equally considers the correct predictions of both classes. Additionally, since there is no significant class imbalance, the potential issues of overemphasizing either precision or recall (which the F1-score addresses) are minimized.

Accuracy Comparison of Models on Train-Validation Phase

Prediction results are obtained from the test set and classification report results are given in Table 4.

Experiment results are given in Table 4 and Figure 2 and show that most of the models have good enough accuracy to handle tool wear classification. LightGBM and XGBoost are significantly accurate classifications compared to others.

Ensemble Learning

In ensemble learning, a soft voting classifier is an advanced technique that merges the probabilistic outputs of several machine learning models to determine the final prediction. This classifier makes decisions based on the combined probabilities provided by all the contributing models. The soft voting classifier operates through the following steps:

Base model training: Multiple base classifiers, denoted as $C_1, C_2, .. , C_n$ are independently trained on the same dataset. These classifiers can be homogeneous (same algorithm) or heterogeneous (different algorithms)

Probability Prediction: For given input data x, each classifier C_j produces a predicted probability vector:

$$P_i = [p_{i1}, p_{i2}, .., p_{ij}]$$

where p_{ij} is the predicted probability that belongs to classifier C_j and j is the total predicting class number.

Table 4. Models and their prediction results on train-validation phase

Model	Tool condition	Accuracy	F1_Score	Support
KNN	Unworn	0.9187	0.9149	1929
	Worn		0.9221	2117
Decision Tree	Unworn	0.9867	0.9860	1929
	Worn		0.9872	2117
Random Forest	Unworn	0.9918	0.9914	1929
	Worn		0.9922	2117
LightGBM	Unworn	**0.9973**	**0.9971**	1929
	Worn		**0.9974**	2117
XGBoost	Unworn	0.9936	0.9933	1929
	Worn		0.9939	2117

Table 5. Models and their prediction results on the test phase

Model	Tool condition	Accuracy	F1_Score	Support
KNN	Unworn	0.9203	0.9139	2332
	Worn		0.9259	2725
Random Forest	Unworn	0.9905	0.9897	2332
	Worn		0.9912	2725
LightGBM	Unworn	0.9968	0.9966	2332
	Worn		0.9971	2725
Voting Classifier	Unworn	**0.9970**	**0.9968**	2332
	Worn		**0.9972**	2725

The formula for the soft-voting classifier final decision:

$$\hat{y} = \arg max \sum_{j\ 1}^{m} p_{ij}$$

where $p_{ij} = P_i(\mathrm{C} \mid \mathrm{x})$ is probability for each class C given an input x.

For a classifier task with m models and C classes, each model j outputs a probability distribution $P_i(\mathrm{C} \mid \mathrm{x})$ for each given class C. This approach effectively leverages the strengths and mitigates the weaknesses of individual models, leading to enhanced overall performance.

In this study, KNN, RF, and LightGBM models are utilized as constituent models for the soft voting classifier. KNN is a non-parametric method that classifies a sample by looking at the predominant class among its nearest neighbors. RF is an ensemble approach that utilizes a collection of decision trees to boost predictive accuracy and prevent overfitting by averaging the predictions from several trees. LightGBM is a gradient-boosting framework that utilizes tree-based algorithms, renowned for its efficiency and outstanding performance. To improve performance, 3 different substructures of the machine learning model were selected. The combined use of these diverse models in a soft voting classifier resulted in an exceptional performance on the train-validation phase, achieving an accuracy of 0.9953, an F1 score of 0.9955, and an ROC AUC of 0.9996. Even though the voting classifier model could not reach the training-validation phase, it shows that the voting classifier model is not as effective as LightGBM (0.9972). We need to evaluate the test data to make a final decision. To handle this, all models, including the voting classifier, are saved to make predictions on the test set, which is taken during the data pre-processing phase.

Test results are given in Table 5 and show that even if the model achieves good results in the training-validation phase, it may lose performance on test data that it has not yet encountered. While gradient-based models, like LightGBM, often demonstrate strong performance during the training and validation phases, there is a possibility that their effectiveness may diminish when applied to new, unseen test data. This phenomenon can arise due to the model's reliance on specific patterns learned from the training data, which may not generalize well to other datasets. In contrast, non-parametric models such as K-Nearest Neighbors (KNN) tend to excel in these situations. KNN operates on the principle of proximity, making predictions based on the closest training data points to a given test instance. This flexibility allows KNN to adapt to the underlying structure of the data more effectively, often resulting in improved performance when faced with previously unencountered data. Thus, as can be seen from Table 5, while both model types have their strengths and weaknesses, the voting classifier can eliminate model weaknesses if it contains diverse types of machine learning models.

CONCLUSION

This research highlights the ability of machine learning algorithms to accurately predict tool wear in machining operations. By utilizing aggregated dataset we systematically explored the effectiveness of machine learning and ensemble learning techniques for tool wear classification in machining operations. Through rigorous experimentation with multiple algorithms—K-Nearest Neighbors (KNN), Decision Trees, Random Forests, LightGBM, and XGBoost—we demonstrated that advanced learning methods can accurately differentiate between worn and unworn tool states. LightGBM and XGBoost emerged as the leading individual models, achieving superior classification performance on training phase. Furthermore, by employing a soft voting ensemble composed of KNN, Random Forest, and LightGBM, we achieved an exceptional test accuracy of 99.70% on unseen test data, underscoring the robustness and reliability of ensemble strategies in industrial predictive maintenance contexts.

These findings highlight the potential of machine learning to enhance tool monitoring, allowing manufacturers to implement proactive maintenance strategies. By improving prediction accuracy, companies can reduce costs associated with tool replacement and improve production efficiency.

Future research may focus on integrating real-time data with different types of materials and exploring additional algorithms to further enhance predictive capabilities with fewer features. Overall, this study provides a promising framework for leveraging advanced analytics in manufacturing to optimize operational performance.

Data Availability Statement

The underlying data repository is publicly available on kaggle [7].

Author's Contributions

Muhammet Ali Aykanat: Conception, Design, Supervision, Data Processing, Analysis, Interpretation, Literature Review, Writer.

Rifat Kurban: Conception, Design, Supervision, Interpretation, Writer, Critical Review.

Conflict of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Statement on the Use of Artificial Intelligence

During the preparation of this work the author(s) used ChatGPT-4 in order to improve grammar and style. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Code Availability

Code is publicly available and link is provided in reference [23].

Ethics

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Ni, J., Liu, X., Meng, Z., & Cui, Y. (2023). Identification of tool wear based on infographics and a double-attention network. *Machines, 11*(10), Article 927. [CrossRef]

[2] Mahardika, M., Taha, Z., Suharto, D., Mitsui, K., & Aoyama, H. (2006). Sensor fusion strategy in the monitoring of cutting tool wear. *Key Engineering Materials, 306*, 727-732. [CrossRef]

[3] Kadhim, K. J., Abbas, A. A., & Hussein, M. A. H. (2018). Effect Ti/AlTiN multilayer coating on the crater wear process of cutting tool and tribological properties. *Al-Khwarizmi Engineering Journal, 13*(4), 58-68. [CrossRef]

[4] Kothuru, A., Nooka, S. P., & Liu, R. (2017). Cutting process monitoring system using audible sound signals and machine learning techniques: An application to end milling. In *Proceedings of the International Manufacturing Science and Engineering Conference, 3*, V003T04A050. [CrossRef]

[5] Terrazas, G., Martínez-Arellano, G., Benardos, P., & Ratchev, S. (2018). Online tool wear classification during dry machining using real time cutting force measurements and a CNN approach. *Journal of Manufacturing and Materials Processing, 2*(4), Article 72. [CrossRef]

[6] Kumar, A. S., Agarwal, A., Jansari, V. G., Desai, K. A., Chattopadhyay, C., & Mears, L. (2023). Vision-based tool wear classification during end-milling of Inconel 718 using a pre-trained convolutional neural network. In *ASME International Mechanical Engineering Congress and Exposition, 3*, V003T03A016. [CrossRef]

[7] Sun, S. (2018). CNC mill tool wear. Available at: https://www.kaggle.com/datasets/shasun/tool-wear-detection-in-cnc-mill Accessed on Oct 3, 2025.

[8] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21-27. [CrossRef]

[9] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81-106. [CrossRef]

[10] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. [CrossRef]

[11] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q; & Liu, T. Y. (2017). *Lightgbm: A highly efficient gradient boosting decision tree*. Advances in Neural Information Processing Systems 30. Curran Associates Inc Publisher.

[12] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785*-794. [CrossRef]

[13] Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia, 4*(2), 1883. [CrossRef]

[14] Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics, 21*(3), 660-674. [CrossRef]

[15] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news, 2*(3), 18 –22.

[16] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems* (pp. 1-15). Springer Berlin Heidelberg. [CrossRef]

[17] Zhou, Z.-H. (2012). *Ensemble methods: Foundations and algorithms* (Vol. 14). [CrossRef]

[18] Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review, 33*(1), 1-39. [CrossRef]

[19] Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine, 6*(3), 21-45. [CrossRef]

[20] Kuncheva, L. (2014). *Combining pattern classifiers: Methods and algorithms: Second edition* (Vol. 47). [CrossRef]

[21] Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research, 11*, 169-198. [CrossRef]

[22] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer. [CrossRef]

[23] Aykanat, M. A., & Kurban, R. CNC-tool-wear-detection. Available at: https://github.com/MAAykanat/CNC-Tool-Wear-Detection Accessed on Oct 03, 2025.