TOOL WEAR STATE CLASSIFICATION USING ENSEMBLE LEARNING AND DEEP LEARNING

SUBMITTED BY
Tejas Kajale
Vivek Jadhav

Abstract:

This project focuses on the accurate assessment and classification of tool wear in manufacturing and machining processes using advanced deep learning and machine learning techniques. The aim is to develop a reliable system for evaluating tool wear and categorizing tool states, thereby enhancing machining operations.

The project employs various deep learning architectures, including Feed Forward Neural Networks (FNN), and Artificial Neural Networks (ANN). These models analyze images of machining tools to detect patterns and characteristics indicative of different levels of tool wear, such as "Normal wear," Moderate Wear," and "Severe Wear. "This project delves into the application of machine learning algorithms, including Logistic Regression, K-Nearest Neighbors (KNN) and SVM, Use of ensemble learning algorithms such as Random Forest and Gradient Boosting for the classification of tool wear in manufacturing and machining processes.

A substantial dataset containing numerical data of tool wear during different machining techniques. The deep learning models undergo meticulous training with emphasis on data preparation and feature extraction to discern subtle variations in tool wear characteristics. Deep learning techniques, particularly fine-tuning pre-trained models, are leveraged to create highly accurate tool wear classifiers.

The results demonstrate promising outcomes, with the developed models achieving an impressive accuracy rate with the use of SMOTE. This high accuracy underscores the potential of deep learning, machine learning, ensemble learning in estimating and categorizing tool wear. Accurate identification of tool wear not only optimizes tool management and maintenance but also reduces costs and enhances productivity in machining operations.

In summary, this experiment showcases the efficacy of algorithm learning in tool wear assessment and classification with the help SMOTE. By providing timely and precise insights into tool conditions, the described models have the potential to revolutionize the industrial sector by ensuring timely replacement or servicing of tools as necessary. The manufacturing and machining field stands to benefit significantly from reduced downtime, cost savings, and improved machining processes facilitated by these advancements.

Dataset Description:

We used IEEE NUAA_Ideahhouse Tool Wear dataset This dataset is employed for several purposes:

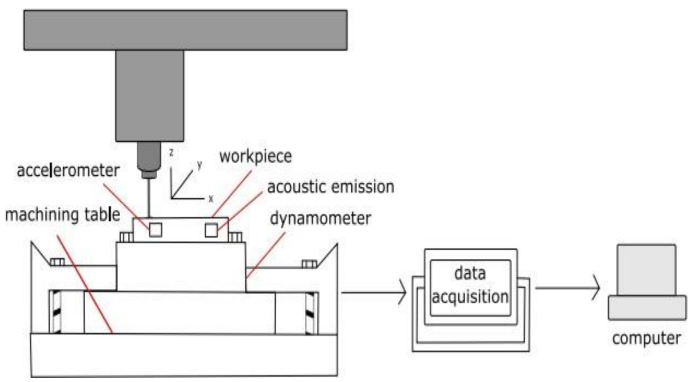
i) Analyzing the impact of process information on monitoring signals under consistent tool wear conditions, utilizing signal processing methods.

ii) Training and testing models for tool monitoring and wear prediction, particularly in scenarios with significant variations in cutting conditions. These variations encompass cutting parameters, the material and geometry of cutting tools, and workpiece materials. The dataset covers cutting conditions characterized by continuous changes, including those found in sidewall machining and closed pocket machining.

Sidewall machining entails a cutting process with fixed cutting conditions, while closed pocket machining involves cutting conditions that continuously vary. This variation occurs because the tool path for closed pockets incorporates line segments, arcs, full cutting, and non-full cutting. Although cutting parameters are fixed in the arc tool path area, the actual cutting parameters, such as feed rate and cutting width, constantly change due to alterations in cutting geometry.

3.3 Data acquisition

The online tool monitoring system is used to collect tool monitoring signals in the machining process, including signals such as cutting force, vibration, spindle current and power. The the monitoring signal acquisition experiment, and the signal acquisition flowchart is shown in the fig below The sampling frequency of each sensor, which are selected considering the cutting speed and spindle speed in our experiments. The low-frequency signals collected from the acquisition software were automatically interpolated during the acquisition.



SMOTE (Synthetic Minority Oversampling Technique):

Tool wear classification, a crucial aspect of machining process optimization, often encounters challenges due to class imbalance in datasets. This imbalance arises when the number of data points representing a specific wear category (e.g., severe wear) is significantly lower compared to others (e.g., normal wear). Consequently, machine learning models trained on such imbalanced data tend to prioritize the majority class, leading to poor performance in classifying the minority class.

Synthetic Minority Oversampling Technique (SMOTE):

SMOTE presents a robust approach to address class imbalance. It specifically targets the minority class and aims to create synthetic data points to achieve a more balanced class distribution. Here's a breakdown of the SMOTE algorithm:

1. **Minority Class Identification:** SMOTE begins by identifying the class with the fewest data points, which often corresponds to the wear category of primary interest (e.g., severe wear).
2. **Nearest Neighbor Selection:** For each data point within the minority class, SMOTE employs a k-Nearest Neighbors (kNN) search algorithm. This identifies k data points in the same class that are closest to the original point in the feature space.
3. **Synthetic Data Generation:** SMOTE leverages the concept of interpolation to create new synthetic data points. It randomly selects a neighbor from the k nearest neighbors identified in step 2. Subsequently, it creates a new synthetic data point by interpolating between the original data point and its chosen neighbor. Essentially, this generates a new point along the line segment connecting these two points in the feature space.
4. **Oversampling with Repetition:** This process of neighbor selection, synthetic data generation, and repetition is performed iteratively for each data point in the minority class. By oversampling the minority class through synthetic data generation, SMOTE aims to achieve a more balanced class distribution within the dataset.

Benefits of Utilizing SMOTE:

- **Enhanced Model Performance:** Addressing class imbalance through SMOTE can significantly improve the performance of machine learning models on the minority class. This translates to more accurate tool wear classification across all categories (normal, moderate, severe).

- **Simplified Model Selection:** With a balanced dataset achieved through SMOTE, the need for complex algorithms specifically designed for handling imbalanced data may be diminished. Standard classification algorithms can often perform effectively on balanced datasets.

By incorporating SMOTE as a data preprocessing step, we significantly enhanced the effectiveness of our tool wear classification model. This leads to more accurate wear predictions, ultimately contributing to improved machining process optimization and quality control.

3. Feature Extraction:

Feature extraction plays a pivotal role in the predictive maintenance of machining tools using machine learning models. It involves transforming raw sensor data collected during machining operations into meaningful numerical representations that capture the underlying patterns and characteristics associated with tool wear progression. In this study, we employed a set of statistical features to extract relevant information from the sensor data. The selected features include:

Mean:

The mean represents the average value of a feature over a specific time interval or window. In the context of machine operations, calculating the mean of sensor readings such as vibration amplitude, temperature, or cutting force provides insight into the baseline behavior of the machining process. Deviations from the mean value can indicate changes in tool condition, such as gradual wear or abnormal tool behavior.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Standard Deviation:

The standard deviation measures the dispersion or variability of a feature around its mean value. A higher standard deviation implies greater fluctuations or instability in the sensor readings, which may be indicative of tool wear-related anomalies or irregularities in the machining process. Monitoring changes in the standard deviation over time can help detect early signs of tool degradation or deterioration.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}$$

Kurtosis:

Kurtosis quantifies the peakedness or flatness of the probability distribution of a feature. High kurtosis values indicate a sharper, more peaked distribution, while low kurtosis values suggest a flatter distribution. In the context of machining tool wear prediction, kurtosis can provide information about the shape and symmetry of sensor data distributions. Changes in kurtosis may reflect shifts in the underlying patterns of tool wear progression, such as increased variability or asymmetry in sensor readings.

$$\text{Kurtosis} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^4}{\sigma^4}$$

Skewness:

Skewness measures the degree of asymmetry in the distribution of a feature around its mean value. Positive skewness indicates a longer tail on the right side of the distribution, while negative skewness indicates a longer tail on the left side.

Skewness analysis helps identify non-normal patterns or trends in the sensor data, which may be associated with specific stages of tool wear or machining conditions. Monitoring changes in skewness values can aid in detecting abnormalities or deviations from expected machining process behavior.

$$\text{Skewness} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^3}{\sigma^3}$$

By extracting these statistical features from the raw sensor data, we aim to capture essential information about the dynamic behavior of machining processes and the evolving condition of machining tools. These features serve as input variables for training machine learning models to predict tool wear and facilitate timely maintenance interventions to prevent unexpected machine failures and optimize machining productivity.

5. Data Preprocessing:

Data preprocessing is a critical phase in machine learning and deep learning pipeline, ensuring that the data is appropriately formatted and prepared for model training. In this study, we performed comprehensive preprocessing steps on the dataset comprising 30 CSV files, each containing 8 parameters and approximately each csv file containing 30,000 rows of sensor data. The objective of data preprocessing was to transform the raw sensor data into a structured format suitable for feature extraction and subsequent model training. The following steps were undertaken:

5.1 Data Classification and Labeling:

- Initially, the sensor data from each CSV file was classified into three distinct categories based on the level of tool wear observed: normal wear, moderate wear, and severe wear. This classification was conducted by domain experts with expertise in machining processes.
- A new column named "target" was added to each CSV file, containing the respective tool wear category label for each row of data. This labeling process facilitated supervised learning, enabling the machine learning models to learn the patterns associated with different levels of tool wear.
- For each CSV file, we extracted statistical features from the sensor data to capture essential characteristics related to tool wear progression. Specifically, we computed the mean, standard deviation, skewness, and kurtosis of each parameter over a sliding window of 300 rows.
- The sliding window approach ensured that the feature extraction process considered the temporal aspect of the sensor data, capturing trends and variations over consecutive data points.

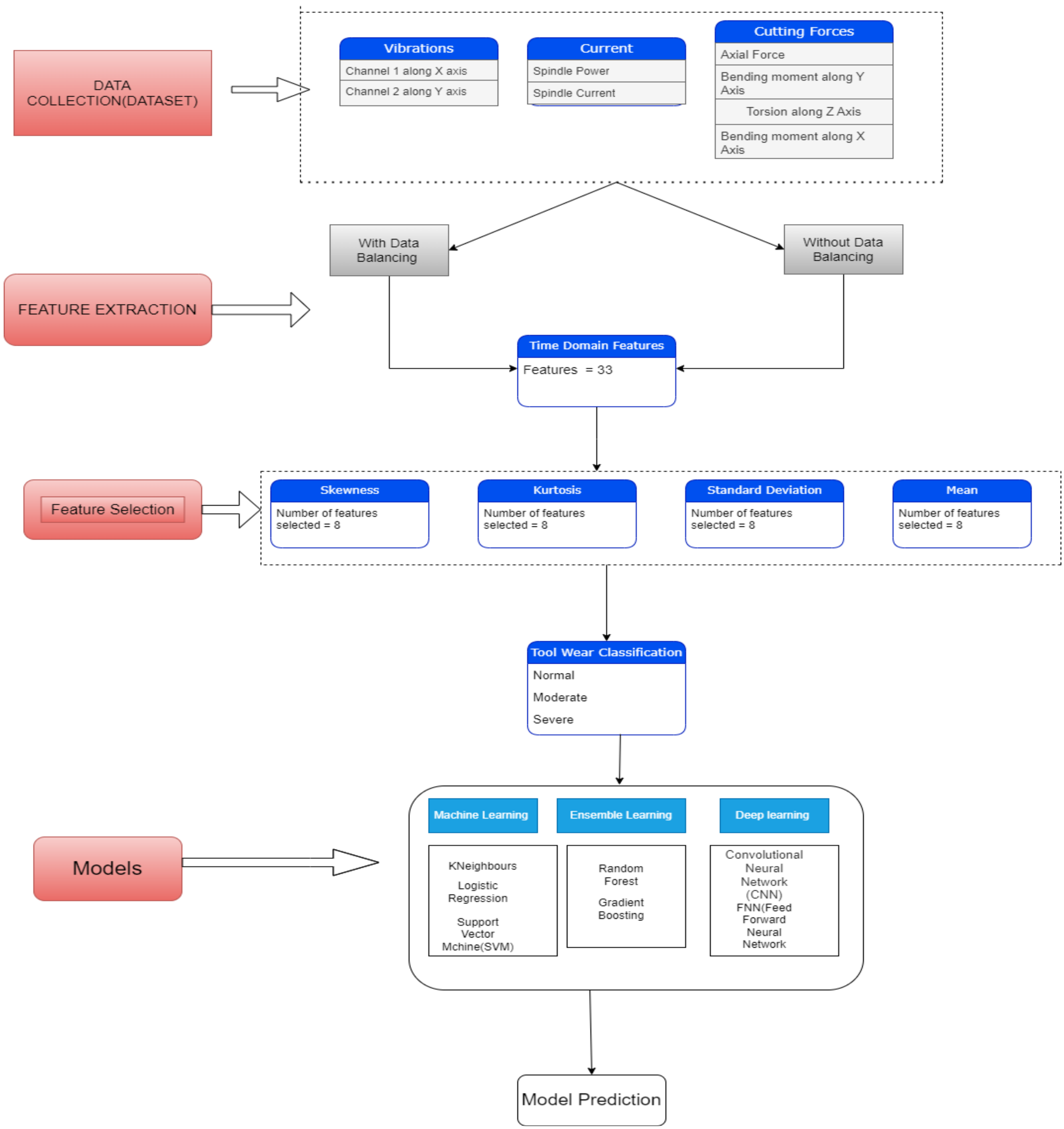
5.2 Data Aggregation and Merging:

- After feature extraction, the statistical features extracted from each CSV file were aggregated into a unified dataset. This involved merging the individual CSV files into a single comprehensive CSV file, where each row represented a unique instance of the machining process, and each column represented a feature extracted from the sensor data.
- The "target" column containing the tool wear category labels was retained in the merged dataset, preserving the association between the features and the corresponding target labels.

5.4 Data Splitting for Training and Evaluation:

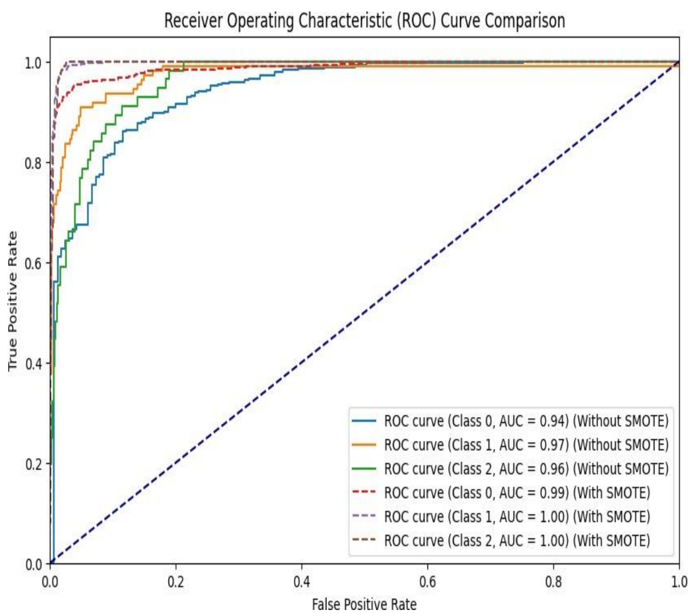
- Finally, the preprocessed dataset was split into training and evaluation subsets to facilitate model training, validation, and testing. The training subset was used to train the Machine learning , Deep learning models, Ensemble learning, while the evaluation subset was reserved for assessing the performance of the trained models.

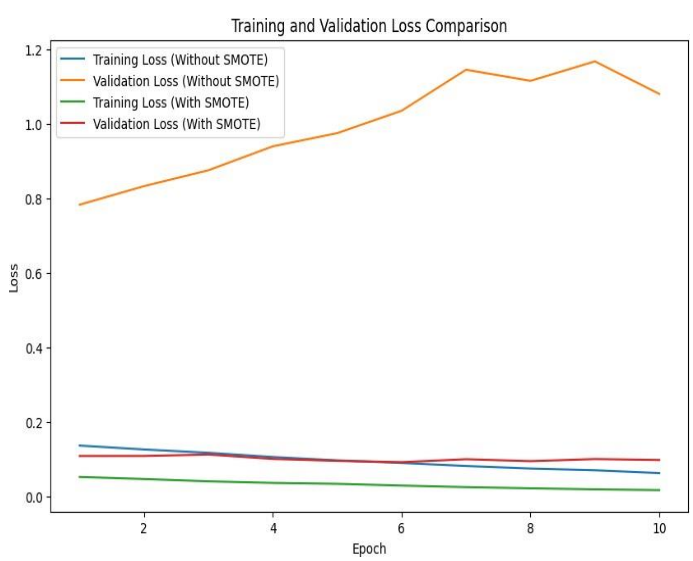
By preprocessing the raw sensor data and transforming it into a structured dataset suitable for learning, we ensured the integrity and reliability of the data for subsequent model training and evaluation.



FNN [Feed Forward Neural Network]:

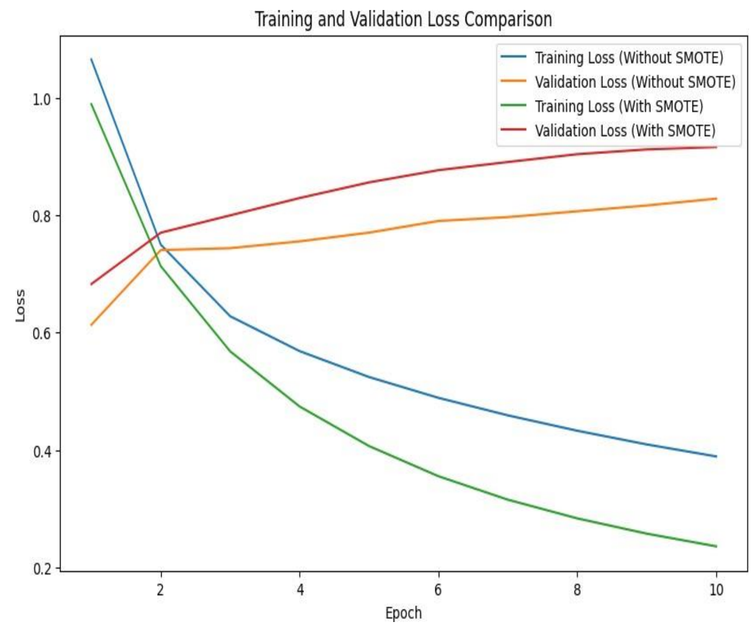
Feed Forward Neural Networks (FNNs) process information in one direction, from input to output, through layers of nodes. Each node applies weights to its inputs, applies an activation function, and passes the result to the next layer. During training, weights are adjusted to minimize the difference between predicted and actual outputs. FNNs are basic but powerful models used for tasks like classification and regression.

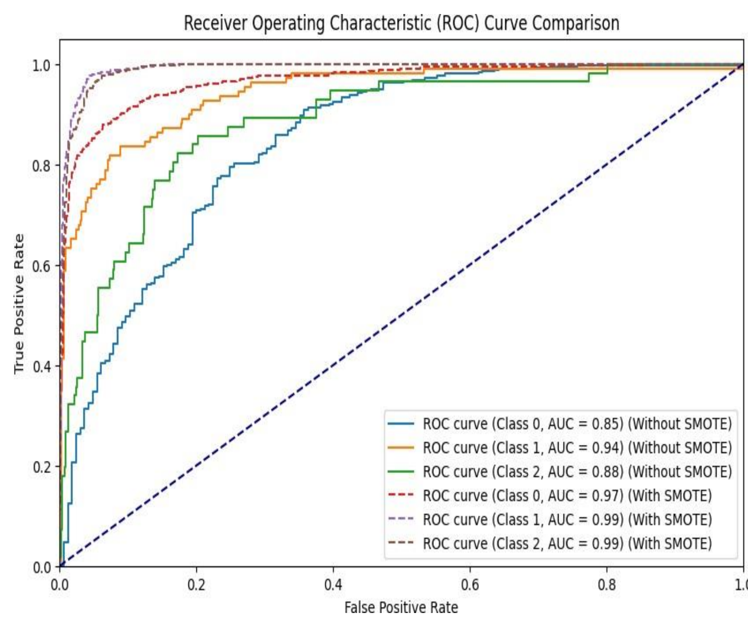




Multilayer Perceptron(MLP):

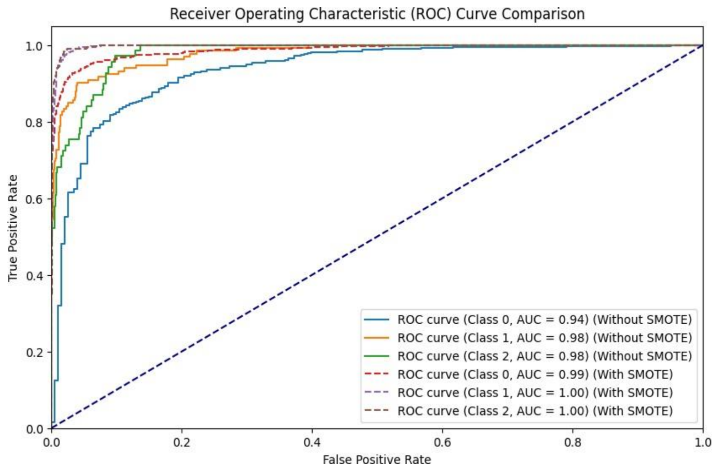
A type of neural network with layers of interconnected nodes. They process data through input, hidden, and output layers, using activation functions to introduce nonlinearity. MLPs are trained using algorithms like backpropagation to minimize prediction errors. They're versatile for tasks like classification an

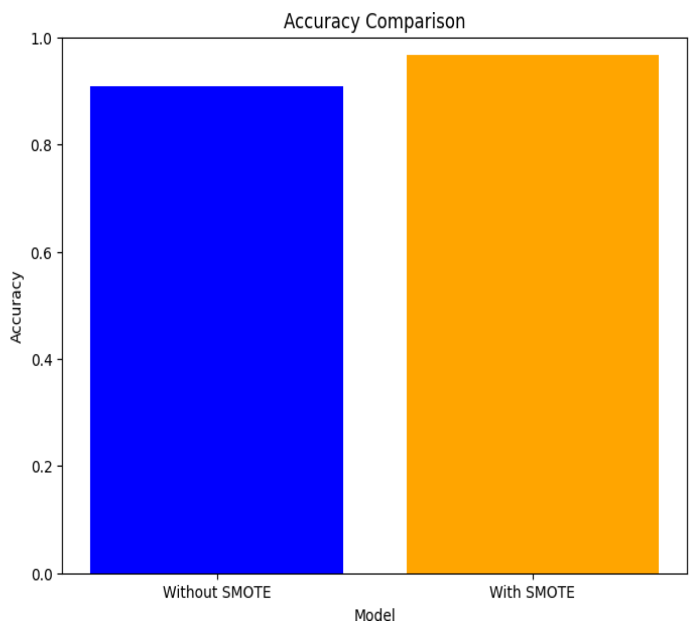




Gradient Boosting:

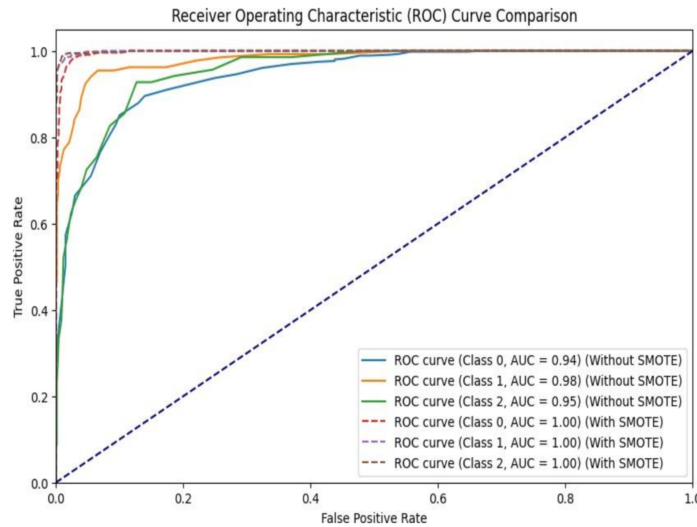
An ensemble learning technique that builds a series of decision trees sequentially. Each tree corrects the errors made by the previous ones, gradually improving the model's predictions. It's effective for both regression and classification tasks and is known for its high accuracy and ability to handle complex relationships in data.

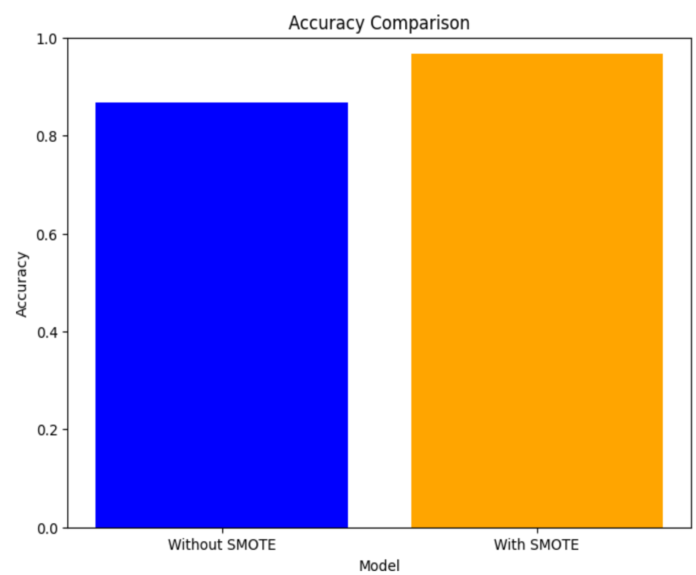




Random Forest:

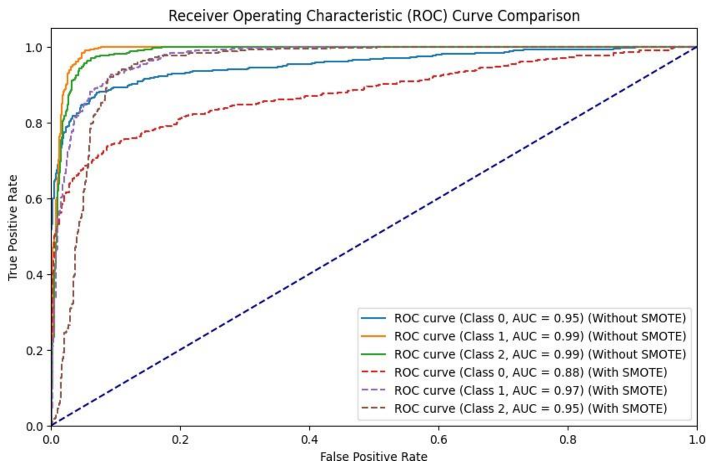
An ensemble learning method that builds multiple decision trees during training. Each tree is trained on a random subset of the data and uses a random subset of features for splitting. The final prediction is determined by averaging (for regression) or voting (for classification) the predictions of individual trees. Random Forest is known for its simplicity, scalability, and ability to handle high-dimensional and noisy features effectively.

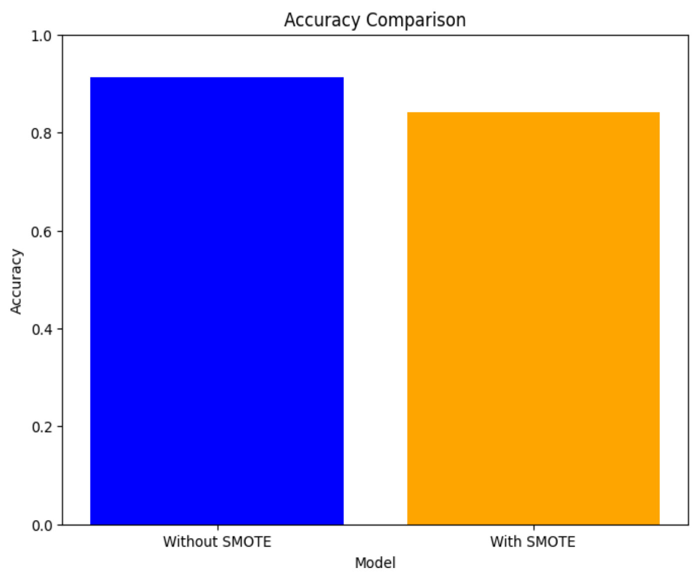




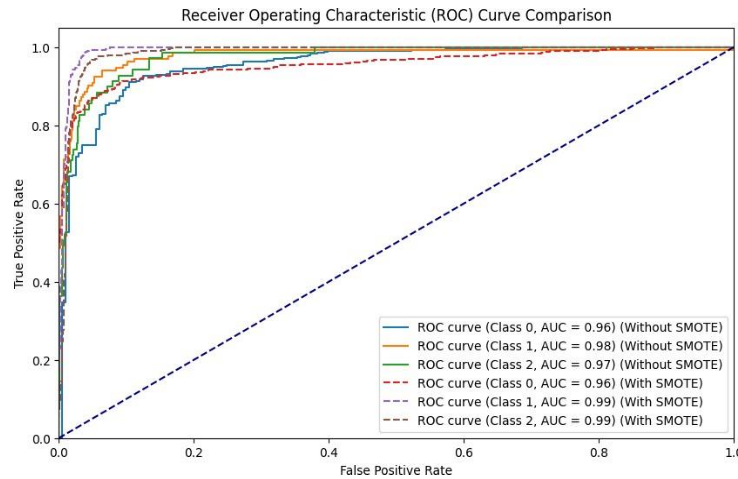
Logistic Regression:

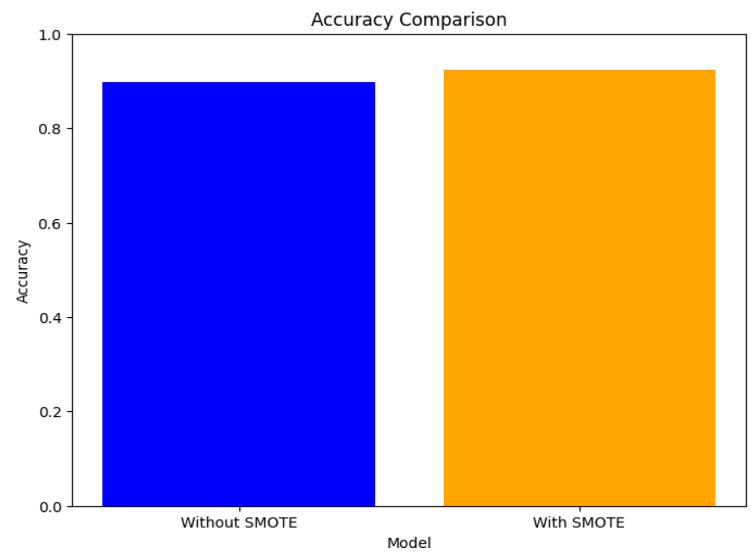
A simple yet effective method for classification. It models the probability of the target class using a linear combination of input features, transformed through the logistic (sigmoid) function. During training, it learns the optimal weights that minimize logistic loss or maximize the likelihood of the observed data. Logistic Regression provides interpretable coefficients and is widely used for its simplicity, efficiency, and interpretability.





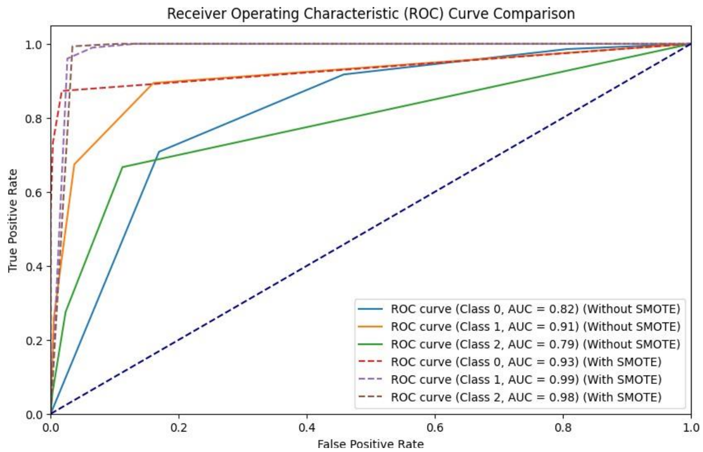
SVM (Support Vector Machines):
A versatile algorithm for classification and regression tasks. It finds the best hyperplane that separates different classes while maximizing the margin between them. SVM can handle linear and non-linear data using kernel functions, and it's effective for high-dimensional data. It's known for its robustness and ability to avoid overfitting, but it can be computationally expensive for large datasets.

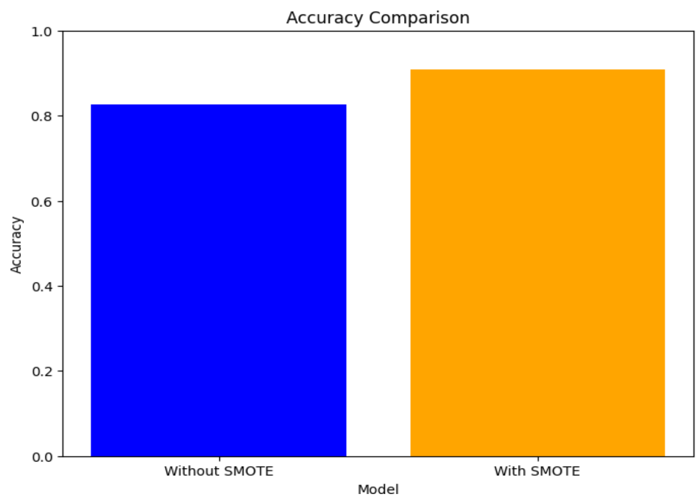




KNN

A simple yet powerful algorithm that classifies data points based on the majority class of their nearest neighbors. It's easy to understand and implement but can be computationally intensive for large datasets. It's non-parametric and doesn't make assumptions about the data distribution, making it suitable for various applications. However, it's sensitive to the choice of the number of neighbors (K) and may struggle with high-dimensional data or noisy features.





Results:

No.	Model	Accuracy Without Data Balancing	Accuracy With Data Balancing
1	Logistic Regression	88.3 %	84%
2	Support Vector Machine (SVM)	89 %	92.6 %
3	KNN	82 %	90%
4	Gradient Boosting	90 %	95.4 %
5	Random Forest	86.4 %	96%
6	Feed Forward Neural Network(FNN)	92 %	97 %
7	(MLP)Multilayer Perceptron	82 %	91.6 %

The study evaluated various algorithms on a dataset, comparing their performance with and without Synthetic Minority Over-sampling Technique (SMOTE). Random Forest and Gradient Boosting achieved high accuracies both with and without SMOTE, indicating robustness to class imbalance. Logistic Regression showed decreased performance with SMOTE, possibly due to its linear nature. KNN and SVM demonstrated notable improvements with SMOTE, suggesting sensitivity to class imbalance. Additionally, FNN and MLP exhibited strong accuracies, particularly with SMOTE, highlighting their effectiveness in handling imbalanced data. Overall, SMOTE enhanced the performance of most algorithms, showcasing its utility in improving classification accuracy, especially for methods sensitive to class distribution.

Conclusions:

The efficiency of machine learning techniques for tool wear classification, focusing on ensemble and deep learning approaches. The employed models exhibited good baseline accuracies exceeding 80% even without the implementation of SMOTE (Synthetic Minority Oversampling Technique). However, a significant improvement in performance (up to a 10% increase in accuracy) was observed when SMOTE was incorporated. These findings strongly suggest that the initial datasets likely suffered from class imbalance, where specific wear categories, particularly severe wear, were underrepresented compared to others like normal wear.

The results highlight the effectiveness of SMOTE in mitigating class imbalance and its subsequent positive impact on model performance. By oversampling the minority class through synthetic data generation, SMOTE allows models to learn from a more balanced representation of wear categories. This enhanced learning translates to improved generalization and classification accuracy across all wear classifications (normal, moderate, severe).

Reference:

1. MacDougall W (2014) Industrie 4.0 Smart Manufacturing for the Future. GTAI Germany Trade and Invest
2. Wang J, Ma Y, Zhang L, Gao RX, Wu D (2018) Deep learning for smart manufacturing: Methods and applications. J Manuf Syst 48:144–156. https://doi.org/10.1016/j.jmsy.2018.01.003, special Issue on Smart Manufacturing
3. Bonifacio M, Diniz A (1994) Correlating tool wear, tool life, surface roughness and tool vibration in finish turning with coated carbide tools. Wear 173(1):137–144. https://doi.org/10.1016/0043-1648(94)90266-6
4. Ambhore N, Kamble D, Chinchanikar S, Wayal V (2015) Tool condition monitoring system: A review. Mater Today: Proc 2(4):3419–3428. https://doi.org/10.1016/j.matpr.2015.07.317, 4th International Conference on Materials Processing and Characterzation
5. Kong D, Chen Y, Li N (2017) Force-based tool wear estimation for milling process using gaussian mixture hidden markov models. Int J Adv Manuf Technol 92(5):2853–2865. https://doi.org/10.1007/s00170-017-0367-1
6. Niaki FA, Ulutan D, Mears L (2015) In-process tool flank wear estimation in machining gamma-prime strengthened alloys using kalman filter. Procedia Manuf 1:696–707. https://doi.org/10.1016/j.promfg.2015.09.018, 43rd North American Manufacturing Research Conference, NAMRC 43, 8-12 June 2015, UNC Charlotte, North Carolina, United States
7. Wang P, Gao RX (2015) Adaptive resampling-based particle filtering for tool life prediction. J Manuf Syst 37:528–534. https://doi.org/10.1016/j.jmsy.2015.04.006
8. Cosme LB, D'Angelo MFSV, Caminhas WM, Yin S, Palhares RM (2018) A novel fault prognostic approach based on particle filters and differential evolution. Appl Intell 48(4):834–853. https://doi.org/10.1007/s10489-017-1013-1
9. Wu D, Jennings C, Terpenny J, Kumara S (2016) Cloud-based machine learning for predictive analytics: Tool wear prediction in milling. In: 2016 IEEE International Conference on Big Data (Big Data), pp 2062–2069. https://doi.org/10.1109/BigData.2016.7840831

10. Sick B (2002) On-line and indirect tool wear monitoring in
turning with artificial neural networks: a review of more than
a decade of research. Mech Syst Signal Process 16(4):487–546.
https://doi.org/10.1006/mssp.2001.1460
11. Wuest T, Weimer D, Irgens C, Thoben KD (2016) Machine learning
in manufacturing: advantages, challenges, and applications.
Prod Manuf Res 4(1):23–45.
https://doi.org/10.1080/21693277.
2016.1192517
12. Terrazas G, Mart´ınez-Arellano G, Benardos P, Ratchev S (2018)
Online tool wear classification during dry machining using real
time cutting force measurements and a cnn approach. J Manuf
Mater Process 2(4):72.
https://doi.org/10.3390/jmmp2040072
13. PHMSociety (2010) 2010 phm society conference data challenge,
https://www.phmsociety.org/competition/phm/10, Accessed January
31, 2018
14. Cui X, Zhao J, Dong Y (2013) The effects of cutting parameters
on tool life and wear mechanisms of cbn tool in high-speed face
milling of hardened steel. Int J Adv Manuf Technol 66(5):955–
964. https://doi.org/10.1007/s00170-012-4380-0
15. Taylor F (1907) On the art of cutting metals. Trans Am Soc Mech
Eng 38:31–35
16. Poulachon G, Moisan A, Jawahir I (2001) Tool-wear mechanisms
in hard turning with polycrystalline cubic boron nitride tools.Wear
250(1):576–586. https://doi.org/10.1016/S0043-1648(01)00609-3,
13th International Conference on Wear of Materials
17. Karandikar JM, Abbas AE, Schmitz TL (2013) Tool life prediction
using random walk bayesian updating. Mach Sci Technol
17(3):410–442.
https://doi.org/10.1080/10910344.2013.806103
18. Sun J, Rahman M, Wong Y, Hong G (2004) Multiclassification
of tool wear with support vector machine by manufacturing
loss consideration. Int J Mach Tools Manuf 44(11):1179–1187.
https://doi.org/10.1016/j.ijmachtools.2004.04.003