XGBoost and Scheduling Algorithm Application for Reduction Maintenance Frequency in Manufacturing 4.0

Keywords: Predictive maintenance, Optimization algorithm, Industry 4.0, Manufacturing Industry

People: Claudio Sasseron, Mona Taghavi

OBJECTIVE

The objective of the study is to develop effective strategies for optimizing maintenance, aiming to reduce the frequency of corrective actions and, consequently, minimize part wastage and maximize tooling usage. The study utilizes classification algorithms such as XGBoost, combined with schedule optimization techniques, to achieve these objectives.

RESEARCH QUESTION

- How can predictive maintenance strategies be optimized to minimize equipment failures?
- What algorithm can be employed to predict machine failures before they occur?"
- How can predictive models for tool wear be utilized to strategically plan maintenance activities?
- How can proactive measures be implemented to minimize waste by avoiding unplanned breaks?
- Can algorithm be used to improve planning production and improve the maintenance actions schedule?

ABSTRACT

In the Industry 4.0 era, the convergence of IoT, Big Data, and AI has paved the way for intelligent production, placing an emphasis on predictive maintenance for improve manufacturing efficiency. This research explores the application of the XGBoost algorithm and schedule optimization techniques for predictive modeling in manufacturing, with a specific focus on minimizing maintenance frequency and extending the lifetime of machine tooling. This research highlights the practicality of advanced machine learning, particularly classification and scheduling distribution, addressing manufacturing challenges and enhancing operational efficiency. Results demonstrate 80% reduction in total scrap and a significant 117% increase in tool usage. The study contributes valuable insights to the optimization of manufacturing processes, showcasing tangible improvements through the implementation of cutting-edge technologies.

INTRODUCTION

In the current economic context marked by globalization and increasingly demanding markets, industries are driven to improve the performance and efficiency of their production lines to strengthen their competitiveness and satisfy their customers. The technologies of the Internet of Things (IoT) and Big Data, and the integration of artificial intelligence (AI) methods play an important role in this context by introducing cognitive automation and, consequently, implementing the concept of intelligent production, leading to intelligent products and services.

This innovative approach leads companies to meet the challenges of a much more dynamic environment. In this regard, companies are able to maximize the life of their equipment while avoiding unplanned downtime and minimizing energy consumption and costs through the application of predictive maintenance [3]. Predictive maintenance, crucial for handling the complexity of interactions within large manufacturing ecosystems, involves predicting equipment remaining life through data collected by various sensors.[5]

It has reached critical importance for industries due to the increasing complexity of interactions between different production activities in increasingly large manufacturing ecosystems.

Figure 1 shows the global trend towards the application of this type of smart maintenance with a prediction of 2022 to 2030.

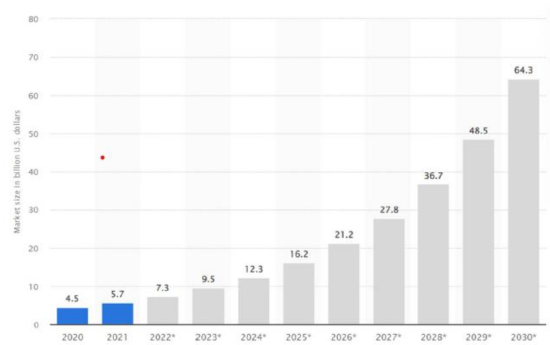

Figure 1 - Size of Maintenance market worldwide in 2020 and 2021 with forecast for future years (2020 to 2030) [6]

In general, the contemporary maintenance approaches vary depending on the different learning models used and the different problems encountered by the machines/equipment.

Modern machinery systems, such as aircraft, naval ships, and manufacturing systems, rely on interdependent machines and components. Effective maintenance is crucial for their efficient operation in terms of cost, availability, and safety. Common maintenance policies include corrective maintenance after failure and preventive maintenance at predefined intervals. Condition based maintenance, a newer technique, has gained attention for its potential to balance maintenance costs and failure costs. It involves continuous real-time monitoring of machines using sensors, not only for condition monitoring and failure diagnostics but also for predictions. Predictions estimate the time of a future failure, known as remaining useful life, or failure probability, by forecasting observed signals. The application of thresholds to failure probabilities and monitored physical parameters, such as temperature, not only saves costs but also enhances productivity, availability, and safety in machinery systems.[35].

Industry 4.0 and Evolution of Industrial Maintenance:

The advent of Industry 4.0 has revolutionized manufacturing by facilitating intercommunication between equipment through IoT, Big Data, computer intelligence, and decision-making systems. This digital transformation allows companies to respond swiftly to market changes, offer personalized products, and increase operational efficiency through informed Big Data. Industry 4.0 devices can autonomously communicate, enabling coordinated operations and real-time information dissemination throughout the production and supply chain [6].

As industries embrace the data-driven connected factory paradigm, maintenance strategies evolve from reactive to preventive and predictive, often referred to as smart maintenance.[23] Preventive maintenance, aimed at preventing failures and extending equipment life, contrasts with reactive maintenance's costly and downtime-intensive nature.[24]

The study explores machine learning applications, specifically XGBoost, in predictive modeling within manufacturing, coupled with schedule optimization techniques. The research investigates a broader range of algorithms, considering XGBoost and schedule optimization for optimizing maintenance frequency in the manufacturing industry.

LITERATURE REVIEW

In the predictive maintenance scenario, the XGBoost algorithm has proven its superiority over classical machine learning approaches by leveraging second-order Taylor expansion on the loss function. This optimization results in a more effective and simultaneously less lost solution, aiming to reduce model variance, simplify the learned model, and mitigate overfitting through the incorporation of a regular term.[33] Proposed by Chen and Guestrin, the XGBoost algorithm stands out as a scalable machine learning system for tree boosting [11]. Notably, it gained significant popularity in the Kaggle 2015 competition, with 17 out of 25 solutions opting for XGBoost.

XGBoost operates by iteratively combining weak base learning models into a stronger learner. At each iteration of gradient boosting, the residual is utilized to correct the previous predictor, optimizing the indicated loss function. The algorithm's parameters and kernel are meticulously defined to ensure optimal performance.The objective function guiding XGBoost's tree boosting model output involves minimizing a complex interplay of training loss functions and penalty terms, signifying the complexity of the algorithm's learning process. The use of second-order Taylor expansion further refines the optimization process, enhancing the algorithm's predictive capabilities[34].

According Fatih Camci[35] the implementation of schedule optimization algorithms in maintenance actions within the manufacturing industry offers a myriad of benefits. It enables the systematic organization of maintenance activities, ensuring that they are strategically planned and efficiently executed. This leads to a reduction in downtime, as maintenance interventions are scheduled during planned breaks, minimizing disruptions to production. Additionally, schedule optimization contributes to the longevity and reliability of machinery by proactively addressing potential issues before they escalate into critical failures.

Schedule optimization algorithm in maintenance actions operate by systematically prioritizing tasks, allocating resources efficiently, and optimizing time for maintenance activities. The algorithm allocates resources effectively, and schedule maintenance during optimal periods to minimize disruption to production. It considers constraints such as machine limitations, budget and regulatory requirements, ensuring feasibility. Some algorithms incorporate dynamic adjustments based on real-time data and iterative learning from completed tasks to refine scheduling decisions over time[29].

By prioritizing and coordinating maintenance tasks, the algorithm assists in minimizing overall equipment downtime and enhancing operational efficiency. Furthermore, it aids in resource allocation by optimizing the utilization of manpower and materials, ultimately resulting in cost savings for the manufacturing enterprise. Overall, the integration of schedule optimization algorithms in maintenance practices proves instrumental in fostering a proactive and streamlined approach to equipment upkeep, positively impacting productivity, reliability, and cost-effectiveness within the manufacturing environment.

Building on the effectiveness showcased in existing literature, this study contributes by exploring XGBoost's potential in predictive modeling for manufacturing environments. The focus is on evaluating the methods within the manufacturing context to build a study case for their implementation. By delving into the intricate relationships of manufacturing data, XGBoost and schedule algorithms hold promise for revolutionizing predictive maintenance in the Industry 4.0 era.

MAINTENANCE FREQUENCY OPTIMIZATION FRAMEWORK

To ensure the effective operation of this framework, establishing a seamless connection between production machines and the dataset is crucial. This process involves configuring both hardware and software systems for data acquisition and specifying interfaces and protocols that enable the real-time transmission of data from the machines.

In this context, it is important to differentiate between dynamic (real-time) and static data. The initial phase of predictive modeling predominantly relies on static data, which is instrumental in constructing predictive models for manufacturing processes. By utilizing static data for model construction, the system can operate seamlessly without disrupting manufacturing processes, guaranteeing consistent efficiency, quality, and enabling.

These models are carefully refined and evaluated for predictive accuracy, processing efficiency, and various performance metrics. Subsequently, these models are deployed to drive data-driven predictive modeling using dynamic, real-time data. Figure 1 offers an illustrative representation of the prediction-optimization concept within the machining processing industry.

Data collection consider various sources, including sensor signals from machines such as torque, rotational speed, process temperature and room temperature, and final machine status. Following this data collection phase, the subsequent step entails a comprehensive data processing, data preparation, classification modeling, optimization, and the identification of optimal scheduling.

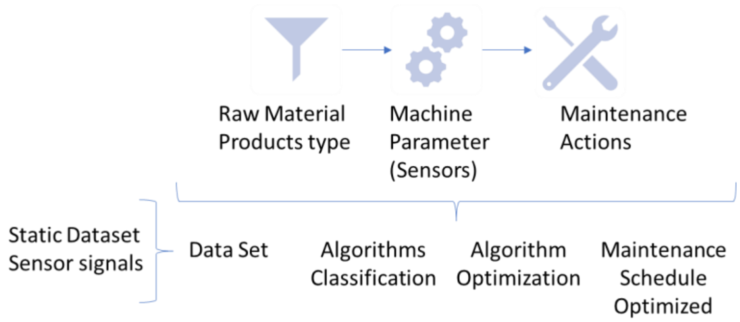

Figure 2- Maintenance Frequency Optimization Framework illustration

The feature engineering is particularly relevant in the realm of machine learning classification. It plays a vital role in extracting relevant data attributes, effectively capturing the intricacies of the machining process.

Hyperparameter tuning plays a crucial role in fine-tuning the model, with techniques such as grid search and random search being deployed to ensure the robustness of the entire process.

Real-time integration with machines is essential, allowing for dynamic adjustments based on optimization results and ensuring a consistent monitoring and control regime to achieve the desired outcomes.

This comprehensive framework represents a structured approach, seamlessly integrating data, classification, and schedule optimization. Continuous monitoring, feedback, and adaptability are essential for long-term success, enhancing machining efficiency, product quality, and predictive maintenance in modern manufacturing.

The pseudocode presented in Figure 2 outlines the algorithm for this framework. It commences with the collection of sensor data, which is then subjected to XGBoost classification to distinguish between approval and failure conditions. Subsequently, a schedule optimizes products sequency to extend tool wear. The system continuously monitors machine operation, and if issues arise, it promptly alerts the maintenance team.

The primary objective is to streamline manufacturing processes, thereby reducing the risk of production interruptions, and increasing tooling lifetime and overall production efficiency.

```
Algorithm ClassificationAndOptimization(Sample S, Features F)
Step 1: Data Collection
  sensor_data = CollectSensorData()

Step 2: Classification Using XGBoost
 good_part_labels = XGBoost_Classification(sensor_data)
 tool_wear_labels = XGBoost_TWF_Classification(parts_produced_data)

Step 3: Schedule Optimization Using Task Optimization
  best_products = TaskOptimizationAlgorithm(sensor_data, good_part_labels,type_of_product)

Step 4: Set Machine Parameters
SetParameters(best_parameters)

Step 5: Machining with Schedule Optimized
while true:
        sensor_data = CollectSensorData()
        if IsMachiningProcess(sensor_data, parameters, best_products):
           // Continue machining
        else:
           MachineOperational(sensor_data)

Function Classification Using XGBoost (sensor_data):
   // Implement XGBoost_Classification for machine parameter classification
   // Return failure classification labels
Function XGBoost_TWF_Classification(parts_produced_data):
   // Implement a XGBoost algorithm for tool wear parameters
   // Return tool wear classification labels
Function AlertMaintenanceTeam():
   // Send an alert to the maintenance team if the machine requires maintenance
Function TaskOptimizationAlgorithm(sensor_data, good_part_labels,type_of_product):
   // Set products sequency for optimization tooling usage
   // Return schedule optimized list
```

Figure 3 - Maintenance Frequency Optimization Framework

METHODOLOGY

In order to validate the proposed framework, an open-source manufacturing process dataset has been utilized as an initial demonstration of its feasibility. It's worth noting that although this dataset originates from the collection of data and information from dynamic manufacturing processes, the models created using this data have not yet been implemented in real-world manufacturing environments for applications such as trend analysis and predictive maintenance.

The machining process and the machining environment used as the case study have not been extensively discussed. However, more details about the machining process and machining environment were explained by Ehmann and Kapoor in [32].

Case Study of CNC Machine Predictive Maintenance Frequency

In the field of machining, CNC often referred to as computer numerical control, involves the automated control of tools through computer programming. This technology is applied to operate various tools including drills, lathes, mills, grinders, routers, and 3D printers. CNC functions by converting a raw material (such as metal, plastic, wood, ceramic, stone, or composite) into a predetermined shape, following meticulously coded instructions. Importantly, this process occurs without the need for a manual operator to directly control the machining operation.

For precise motion control along multiple axes, typically at least two axes (X and Y), along with a tool spindle that operates in the Z-axis (depth). The tool's position is managed by direct-drive stepper motors or servo motors, ensuring highly precise movements. In commercial metalworking machines, closed-loop controls are considered standard and necessary to meet the required levels of accuracy, speed, and repeatability. A representation and the photo of machining are shown in Figure 2.

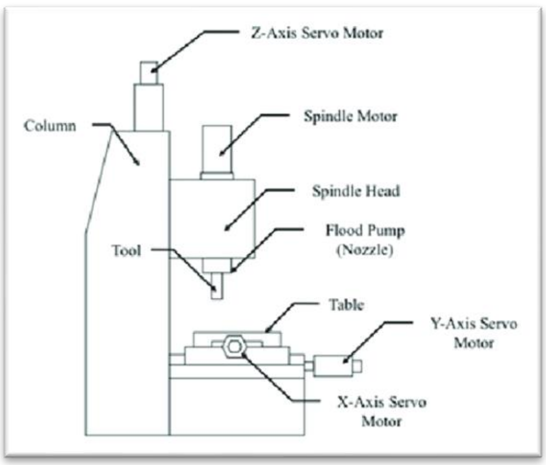

Figure 4 - CNC Machine Representation and photo of machine's tool.

The workflow is illustrated in Figure 3, aligning with the framework outlined in section 2. Initially, the desired product to be produced is categorized as Small, Medium, or High based on client demand. The machine produces the part, and through sensors, it sends parameters of the process to the algorithm. After the part is manufactured, it undergoes quality checks according to predefined requirements, and the algorithms are subsequently fed. Following algorithm classification and optimization, the output determines whether the process should pause for adjustments or proceed to the next part.

Data Flowchart

The presented flowchart, consisting of nine sequential steps, delineates a systematic manufacturing workflow. It begins with Step 1, where products are defined into size categories (Small, Medium, Large). This is followed by Step 2, involving machine production, and Step 3, which assesses the approval status of the manufactured products. If the product is approved (Step 3, "yes"), the process proceeds to Step 4 for the next product; if not (Step 3, "no"), Step 5 initiates operation adjustments. Step 6 involves the generation of various outputs, including Breaks, Tool Wear, Torque, Tool Speed, and Temperature. These outputs undergo analysis in Step 7 through classification models and an optimization model in Step 8. The workflow includes a critical evaluation of maintenance breaks at Step 9, branching to operation adjustment if needed ("yes") or advancing to the next product if not ("no"). This comprehensive approach ensures an organized and optimized manufacturing process, incorporating quality control measures and leveraging classification and optimization models to enhance overall productivity and product quality.

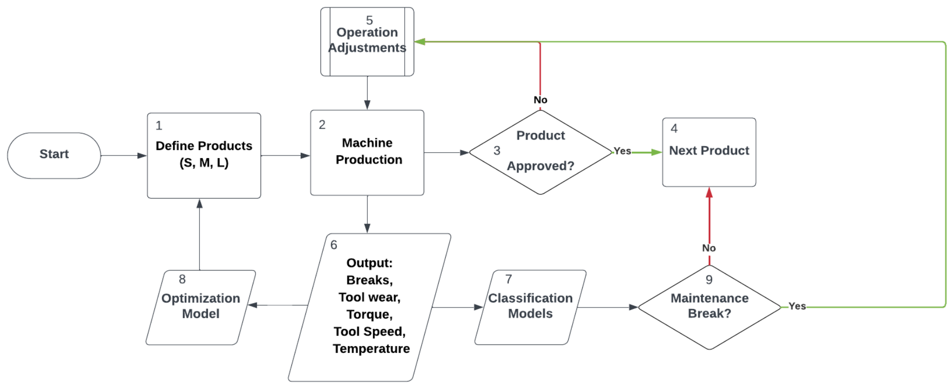

Figure 5- Maintenance Frequency Optimization Model Flowchart

Dataset Handling

The manufacturing process begins with the collection of data, utilizing sensors and maintenance shutdowns to gather relevant information. This raw data is then processed, creating a comprehensive database while addressing challenges like handling noisy data and managing outliers. The next steps involve engineering features, categorization, and interpreting the data to understand its distribution and uncover correlations and relationships. In preparation for building the classification model, the data is further refined. The model-building process includes collecting classified data, preparing the refined data, constructing the model, and applying it.

The results are then checked, and the scheduling phase begins. The schedule optimization involves checking the maintenance frequency, and the finalized schedule is applied to the manufacturing process, ensuring efficient and optimized operation.

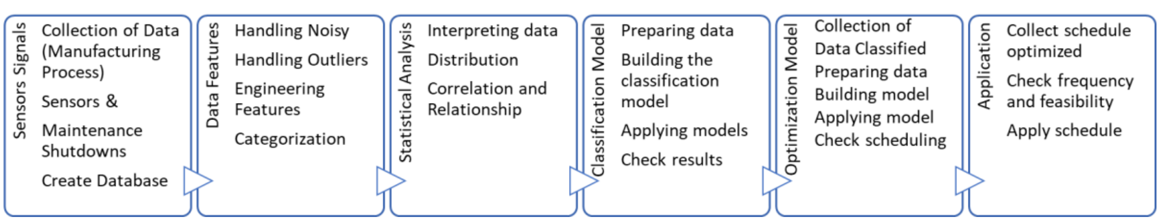

Figure 6 – Representation of data workflow.

Dataset Description

The data used for the experimentation and investigations come from a 10.000 observation run spanning 264 hours. Large quantities of data are available from UC Irvine Repository [33].

Each unique identifier (UID) from 1 to 10,000 is associated with a specific product ID, denoted by the letters L, M, or H, representing quality and time production (finishing) characteristics, with low, medium, and high-quality products taking 2, 3 and 5 minutes respectively.

The process parameters include air temperature, measured through a random walk. Additionally, process temperature, rotational speed, torque values are generated from machine sensor. Tool wear varies based on quality variants, adding 5, 3, or 2 minutes for each variant.

Machine failures are categorized into Tool Wear Failure (TWF), Heat Dissipation Failure (HDF), Power Failure (PWF), Overstrain Failure (OSF), and Random Failures (RNF) with specific conditions and criteria leading to process failure.

The data collected include factory ambient conditions, temperature (K), torque (Nm), tool speed (rpm), tool wear (min) and number of parts as present in table 1.

Table 1 - Data collected in manufacturing environment by sensors and observations.

Features	mean	std	min	50%	max
UDI	5000.5	2886.9	1.0	5000.5	10000.0
Air temperature [K]	300.0	2.0	295.3	300.1	304.5
Process temperature [K]	310.0	1.5	305.7	310.1	313.8
Rotational speed [rpm]	1538.8	179.3	1168.0	1503.0	2886.0
Torque [Nm]	40.0	10.0	3.8	40.1	76.6
Tool wear [min]	108.0	63.7	0.0	108.0	253.0
Machine failure	0.0	0.2	0.0	0.0	1.0
TWF	0.0	0.1	0.0	0.0	1.0
HDF	0.0	0.1	0.0	0.0	1.0
PWF	0.0	0.1	0.0	0.0	1.0
OSF	0.0	0.1	0.0	0.0	1.0
RNF	0.0	0.0	0.0	0.0	1.0

Feature Engineering

In the realm of engineering feature analysis, a suite of formulas plays a pivotal role in quantifying and scrutinizing crucial facets of manufacturing processes. Here, we outline and elucidate each formula, shedding light on their significance and applications.

The Equation (1) quantifies the temperature variation throughout the manufacturing process. Obtained by subtracting the initial temperature from the final temperature, it provides key insights into temperature dynamics during the operation.

$$Temp_{Delta(K)} = Process\ Temperature - Process\ Temperature \ (1)$$

In Equation (2) the formula calculates the energy consumption for the machining process. Involving rotational speed and torque, it offers a measure of the energy required, crucial for evaluating the efficiency of the manufacturing operation.

$$Power(Watts) = 2 * \pi * Rotational\ speed * \frac{Torque}{60} \ (2)$$

Reflecting the time required for processing each part, the Equation (3) aiding in planning and resource allocation based on the complexity of each part, with assigned values for Low (L), Medium (M), and High (H) complexity levels.

$$[L: 2, M: 3, H: 5] = \text{Time\_Part – Time processing for each part} \ (3)$$

The combination of different causes of failures related to process configuration differently than tool wear condition (TWF) presented in Equation (4) are computed by summing various failure types (HDF, PWF, OSF, RNF) and subtracting the total tool wear failures (TWF). This metric provides a consolidated count of unexpected disruptions during the manufacturing process.

$$Failures = \sum (HDF + PWF + OSF + RNF) - \sum TWF \ (4)$$

Planned Stops according to Equation (5) for total breaks excluding failures and tool wear failures. It offers insights into scheduled downtime for maintenance or other planned interruptions, facilitating efficient resource management.

$$Planned\_Stop = \sum Total\ Breaks - \sum (Failures + TWF) \ (5)$$

These formulas collectively establish a robust framework for evaluating and optimizing efficiency, energy consumption, and reliability in manufacturing processes. Their application enables data-driven decision-making, fostering continuous improvement in engineering operations.
The table 2 presents a comprehensive overview of key features in a manufacturing process, providing statistical insights into the central tendencies and variations of critical parameters. Each feature captures a distinct aspect of the manufacturing operation, contributing to a holistic understanding of the process dynamics. The TWF feature indicates the occurrence of tool wear failures, with a binary representation (0 or 1). The data shows that tool wear failures are present 46 cases in total. Temp Delta, has a mean value of 10 K and a standard deviation of 1 K. With a range from 8 K to 12 K. Power consumption in watts, derived from rotational speed and torque, demonstrates a mean of 6280 watts and a standard deviation of 1067 watts. The power values range from 6271 watts to 10470 watts, reflecting the varying energy requirements for the machining process. Time per part, a critical metric for efficiency, has a mean of 3 minutes and a

standard deviation of 1 minute. Ranging from 2 to 5 minutes, this feature provides insights into the time efficiency associated with processing different levels of part complexity.

The Failures feature, indicating the occurrence of various failure types, shows a binary representation (0 or 1). The data reveals that failures are present 293 in total.

Planned Stop represents scheduled stops for maintenance or interruptions, excluding failures. With a binary representation (0 or 1), the data suggests the occurrence in 62 of planned stops in total.

Table 2 - Descriptive dataset after feature engineering review

Features	mean	std	min	50%	max
Air temperature [K]	300	2	295	300	305
Process temperature [K]	310	1	306	310	314
Rotational speed [rpm]	1539	179	1168	1503	2886
Torque [Nm]	40	10	4	40	77
Tool wear [min]	108	64	0	108	253
TWF	0	0	0	0	1
Temp Delta	10	1	8	10	12
Power (watts)	6280	1067	1148	6271	10470
Time/Part	3	1	2	2	5
Failures	0	0	0	0	1
Planned Stop	0	0	0	0	1

In summary, the table encapsulates a wealth of information about the manufacturing process, facilitating in-depth analysis and informed decision-making for optimizing operational efficiency and reliability. The mean, standard deviation, and range metrics offer a nuanced understanding of the variability and central tendencies of each feature, contributing to a comprehensive assessment of the manufacturing system.

Balance Analysis for Failures Cases

The issue of imbalanced samples is further exacerbated when considering the specific types of failures within the dataset. In a total of 10,000 parts, there are 293 instances of diverse failures, which encompass various failure types such as HDF (Hardness Defect Failure), PWF (Porous Walled Failure), OSF (Out of Specification Failure), and RNF (Run out of Specification Failure). However, among these failures, there are only 46 instances categorized specifically as tool wear failures.

This disparity in the distribution of failure types introduces an additional layer of complexity. Not only is the dataset imbalanced in terms of failures and non-failures, but there is also an uneven representation among different failure categories. The low occurrence of tool wear failures compared to the overall diverse failures poses a challenge for the model to effectively learn and generalize patterns related to tool wear issues.

To address this nuanced imbalance, the use of resampling techniques like SMOTE and bootstrapping becomes even more pertinent. These methods can be tailored to focus on specific failure categories, such as tool wear failures, ensuring that the model receives adequate exposure to instances of interest. By strategically augmenting the dataset with synthetic samples and creating diverse resampled datasets through bootstrapping, the machine learning model gains a more comprehensive understanding of the various failure scenarios, including those associated with tool wear. This targeted approach to addressing imbalances enhances the model's capacity to make accurate predictions and informed decisions across the entire spectrum of failure types.

Over-sampling Technique

After applying the Synthetic Minority Over-sampling Technique (SMOTE) to the dataset, the class imbalances have been significantly mitigated. The initial distribution, with 6,932 instances of no failures and 7,337 instances of failures, has been rebalanced to create a more equitable representation of both classes. This rebalancing is crucial for the machine learning model to effectively learn from and predict instances of both failure and non-failure scenarios.

Further, when considering the specific context of tool wear failures (TWF), the application of SMOTE separately for TWF and non-TWF cases has proven advantageous. In the initial dataset, there were 7,764 instances with TWF and 7,391 instances without TWF. By addressing the imbalance separately within these subcategories, the resampling techniques have enabled the creation of synthetic samples tailored to the nuances of each failure type.

Notably, the resampling has been especially impactful for tool wear failures, where the initial dataset had only 46 instances. The resampling process has significantly increased the representation of tool wear failures, allowing the model to better discern patterns associated with this specific failure category. This targeted augmentation of TWF instances ensures that the machine learning model is not only well-equipped to handle the overall imbalance between failure and non-failure cases but also adept at capturing the intricacies within each failure type.

In summary, the application of SMOTE and bootstrapping, with a specific focus on different failure categories, serves as a strategic and effective approach to addressing imbalances in the dataset. By tailoring the resampling techniques to the specific characteristics of failure types, the model gains a more nuanced understanding, enhancing its predictive capabilities across the entire spectrum of failure scenarios, including those related to tool wear.

Table 3 - Total of observation for each case after application increase sampling methods.

Database	Failures (Combined)	TWF	Total
Dataset - Original	293	46	10.000
Generated 'Failures'	7337	--	14.269
Generated 'TWF'	--	7764	15.155

Experimental Setup – Xgboot

The implementation of all machine learning methods utilized the Scikit-learn library [36,54] and the Jupyter Lab environment. Specifically, the investigated algorithms were coded using the library functions available. The hyperparameters or control parameters for these methods were set based on references [33] and the application of GridSearchCV provided by the Scikit-learn library, unless explicitly mentioned otherwise. It is assumed that these settings are unfamiliar to most production engineers who may not have experience in machine learning. It is conventional that machine learning methods perform better with adjusted or optimized hyperparameters, however these aspects are not deeply explored within the scope of this work. The table 4 shows the result of setup after GridSearchCV application for XGBoost algorithms optimazing the accuracy of prediction for ‘failures’ cases.

Table 4 - Setup Parameters after GridSearchCV technique for XGBoost Algorithm for failures prediction

Parameters	Value	Description
N_estimators	2000	
max_depth	3	Control over-fitting
subsample	0.8	control the sample’s proportion
learning_rate	0.05	
objective	Logistic	Minimizing the loss function
'min_child_weight'	5	Defines the minimum sum of weights
'reg_alpha'	0.1	L1 regularization term on weights
'reg_lambda'	0.1	L2 regularization term on weights

The results presented in Table 5 illustrate the outcomes of the setup after applying GridSearchCV to optimize the accuracy of XGBoost algorithms for predicting tool wear cases, with a specific emphasis on tooling lifetime. The application of GridSearchCV reveals differences in the setup for both failure and tool wear predictions, particularly in the choice of objective functions. Logistic regression is applied for 'failures,' while ridge regression is employed for 'TWF’ cases, showcasing the tailored optimization based on the distinct nature of the prediction tasks.

Table 5 - Setup Parameters after GridSearchCV technique for XGBoost Algorithms for Tool Wear (TWF) prediction

Parameters	Value	Description
N_estimators	2000	
max_depth	3	Control over-fitting
subsample	0.8	control the sample’s proportion
learning_rate	0.05	
objective	Ridge	Minimizing the loss function
'min_child_weight'	15	Defines the minimum sum of weights
'reg_alpha'	0.1	L1 regularization term on weights
'reg_lambda'	0.1	L2 regularization term on weights

Experimental Setup – Task Schedule Optimization

The experimentation setup for the task optimization algorithm is designed to simulate real-world manufacturing scenarios by considering each product as a small task and the tool wear as the entire schedule. A schedule duration is defined based on the maximum tool wear expected, set to 200 minutes as a minimum guarantee by the supplier of the tooling. The key configurations involve choosing a sorting strategy for tasks, such as descending order or random shuffling, and exploring diverse experimental scenarios to comprehensively test the algorithm under varying conditions.

To evaluate the algorithm's effectiveness, performance metrics serve as benchmarks, covering aspects like total time consumed, batch details, maintenance stops, machine working time, and parts produced. The experiment execution phase involves running the algorithm for each scenario, recording output and metrics, and conducting multiple runs to account for randomness. Post-experiment analysis focuses on identifying patterns, trends, and potential areas for optimization. Fine-tuning strategies may include experimenting with alternative sorting methods or adjusting parameters to enhance overall performance. A commitment to comprehensive documentation and reproducibility ensures transparency and facilitates further analysis and refinement by other researchers or stakeholders. In summary, this systematic approach to experimentation provides valuable insights into algorithm behavior and lays the groundwork for continuous improvement and adaptability to various manufacturing contexts.

RESULTS AND DISCUSSION

XGBoost Classification

The XGBoost classifier demonstrated outstanding performance across multiple metrics, confirming its reliability for the given classification tasks. During the evaluation for predicting failures, the model achieved an accuracy of 99.58%, indicating a high proportion of correct predictions. Precision for failures was notably high at 99.39%, signifying minimal false positive instances in identifying stop-inducing failures. The model excelled in recall at 99.80%, highlighting its effectiveness in capturing cases where the system should stop, thus minimizing the occurrence of false negatives. The f1-Score, a comprehensive metric balancing precision and recall, achieved an impressive 99.59%, emphasizing the classifier's ability to strike a harmonious equilibrium between identifying failures and minimizing false alarms.

Moreover, the XGBoost model exhibited remarkable performance in predicting TWF (Tool Wear Failure). It achieved an even higher accuracy of 99.71%, showcasing its precision in identifying instances related to tool wear. Precision for TWF soared to 99.93%, reflecting an exceptional ability to accurately pinpoint tool wear failures with minimal false positives. The model maintained a robust recall of 99.87%, emphasizing its efficacy in capturing the majority of tool wear-related instances. The f1-Score for TWF reached an out standing 99.93%, reinforcing the classifier's capacity to deliver accurate and reliable predictions for tool wear failures. In summary, the XGBoost classifier consistently demonstrated superior performance across different classification categories, affirming its effectiveness and versatility in predictive modeling.

Table 6 - Performance Testing for XGBoost for Predicit 'Failures' and 'TWF'

XGBoost	**Accuracy**	**Precision**	**Recall**	**f1-Score**
Failures	0.9958	0.9939	0.9980	0.9959
TWF	0.9971	0.9993	0.9987	0.9993

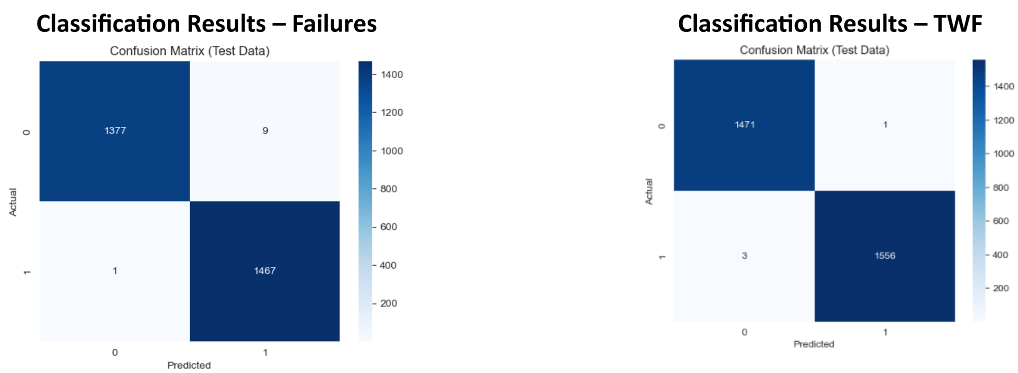

Figure 7- Confusion Matrix showing the Failures and TWF for Actual and Predict cases

Optimization Schedule Production

The task optimization algorithm employed in production manufacturing yielded significant improvements, particularly evident in the remarkable increase in the number of products per schedule. Prior to the optimization implementation, the manufacturing process handled up to 23 products per schedule. However, following the application of the task optimization algorithm, this number surged to an impressive 50 products per schedule. This substantial enhancement underscores the algorithm's effectiveness in streamlining and maximizing production efficiency. The substantial increase in the number of products per schedule can be attributed to the algorithm's ability to intelligently allocate and distribute tasks, resulting in a more optimized and streamlined production process. By systematically organizing and scheduling tasks, the algorithm minimizes idle time and enhances resource utilization, allowing the manufacturing system to accommodate a significantly larger number of products within the same schedule duration. This outcome is indicative of the algorithm's positive impact on production throughput, demonstrating its potential to drive operational efficiency and increase overall manufacturing output. The task optimization algorithm proves instrumental in meeting the demand for higher production volumes, showcasing its applicability and benefits within a production manufacturing setting.

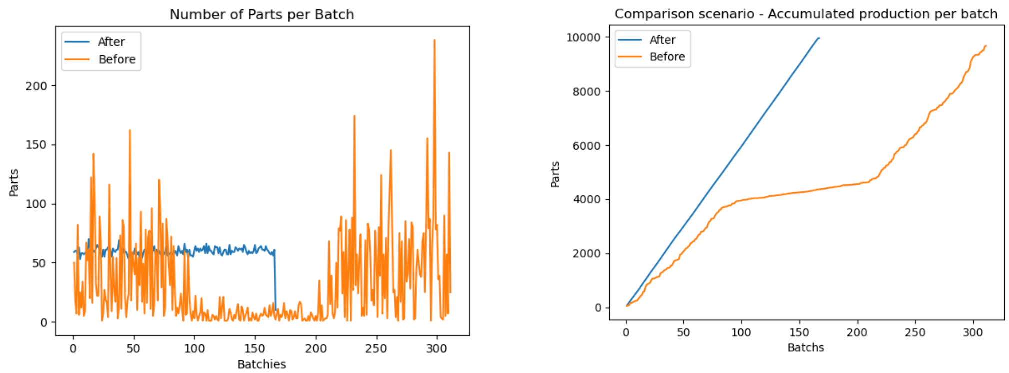

Figure 8 - Graph comparing before and after model application showing the number of parts per batch and accumulation production.

The Table 7 illustrates the improvements in performance measures before and after the implementation of optimizations. The total approved parts, based on 10,000 observations, increased from 9661 to 9933, reflecting a significant improvement of 3.2%. The reduction in total scrapped parts decreasing from 339 to 67, indicating 80% improvement. Similarly, the total number of breaks decreased from 403 to 197, marking a 51% improvement. In terms of efficiency, the average number of parts per tooling, increased from 23 to 50, signifying a remarkable 117% enhancement. These results collectively highlight the effectiveness of the implemented measures in enhancing manufacturing productivity, minimizing waste, and optimizing overall performance.

Table 7 - Table of Measures and Improves results of before and after model application.

Measures	Before	After	Improve (%)
Total Approved Parts (10.000 obs)	9661	9933	3.2
Total Scraped	339	67	80
Total Break	403	197	51
Stop rate ()	0.68	0.11	83
Total waste time (h/day)	1.01	0.43	57
Average Parts per Tool (batch)	23	50	117

CONCLUSION

The study encompasses a comprehensive three-phase project dedicated to harnessing machine learning techniques for predicting and optimizing maintenance actions. The primary focus areas include utilizing XGBoost for predicting failures, employing XGBoost for predicting tool wear failures, and implementing task optimization to strategically organize product sequencing and maximize tool usage. The initial phase entails feature analysis to uncover correlations and engineer relevant features. Following this, classifiers are trained using GridSearchCV and the XGBoost model with top-ranking parameters for both failure detection and tool wear failure prediction. The third phase involves task optimization to efficiently distribute the sequence of products along the tooling lifetime, maximizing parts production before the need for maintenance. The study emphasizes the anti-overfitting capabilities of the proposed ensemble learning method, demonstrating significant improvements in model results when comparing the before and after application phases.

REFERENCES

1. Ledouxa Y., Sébastiana P., Samperb S., Optimization method for stamping tools under reliability constraints using genetic algorithms and finite element simulations. *JMPT 2010, 210, 474-486*.
2. Antunes Rodrigues J., Torres Farinha J., Mendes M., Mateus R. J. G., Marques Cardoso A. J., Comparison of Different Features and Neural Networks for Predicting Industrial Paper Press Condition. *Energies 2022, 15, 6308.*
3. Zandieh M., Fatemi Ghomi S.M.T., Moattar Husseini S.M., An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation 2006, 180, 111–127.*
4. Pinedo M. (Columbia University, New York), Hadavi K. (Siemens Corporate Research, Princeton), Scheduling: Theory, Algorithms and Systems Development. Personal communication, 1992.
5. Chai T., Draxler R.R., Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geosci. Model Dev, 2014, 7, 1247–1250*.
6. Cus F., Zuperl U., *Journal of Materials Processing Technology, 2016, 173, 281–290.*
7. Chinyao L., Chih-Ming H., Fong-Jung Y., Analysis of variations in a multi-variate process
8. using neural networks. *J Adv Manuf Technol 2003, 22, 911–921.*
9. Changchun L., Dunbing T., Haihua Z., Qingwei N., (University of Aeronautics and Astronautics, Nanjing, China) A Novel Predictive Maintenance Method Based on Deep Adversarial Learning in the Intelligent Manufacturing System, 2021.
10. Ehmann K. F., Kapoor S.G., Lazoglu I., Machining Process Modeling: A Review. *Journal of Manufacturing Science and Engineering, 1997, 119, 655–663.*
11. Khiat S., Rahal S. A. H., Senaï B. Classification Predictive Maintenance Using XGboost with Genetic Algorithm. *Revue d'Intelligence Artificielle 2022, 36, 833-845.*
12. Zhang D., Qian L., Mao B., Huang C., Si Y., A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGBoost. *IEEE ACCESS, 2018, 2169-3536.*
13. Camci F., System Maintenance Scheduling with Prognostics Information Using Genetic Algorithm, *IEEE Transactions on Reliability, 2009, 58, no.3.*
14. GitHub, Codes online: https://github.com/Sasseron/Maintenance_Optimization

Appendix A

The Batch distribution organized before and after optimization schedule model application:

```
Batche List Before: [50, 18, 7, 82, 6, 25, 12, 34, 5, 9, 67, 52, 61, 20, 1
22, 16, 142, 100, 32, 22, 22, 89, 68, 1, 7, 27, 19, 16, 4, 116, 39, 9, 55,
27, 17, 54, 3, 12, 73, 11, 86, 80, 20, 4, 18, 24, 162, 18, 59, 49, 40, 66,
10, 54, 31, 93, 17, 49, 7, 78, 16, 72, 77, 11, 96, 5, 14, 61, 58, 18, 120,
95, 29, 83, 5, 12, 87, 70, 49, 31, 72, 10, 64, 26, 5, 12, 7, 14, 24, 10, 1
, 60, 6, 7, 58, 21, 7, 1, 9, 3, 22, 6, 1, 5, 8, 20, 1, 8, 7, 1, 10, 1, 1,
1, 6, 5, 2, 5, 2, 1, 21, 3, 13, 21, 1, 1, 1, 3, 11, 8, 2, 1, 6, 3, 10, 15,
2, 1, 13, 10, 1, 5, 8, 12, 1, 3, 1, 8, 4, 2, 7, 1, 1, 5, 7, 5, 6, 12, 6, 5
, 4, 14, 5, 8, 16, 5, 2, 7, 11, 1, 6, 4, 8, 16, 3, 9, 6, 1, 10, 9, 2, 4, 9
, 3, 3, 14, 17, 15, 1, 2, 3, 1, 1, 5, 1, 8, 6, 3, 1, 5, 1, 7, 35, 1, 14, 1
, 2, 3, 3, 6, 68, 15, 39, 7, 3, 8, 50, 12, 79, 77, 89, 24, 59, 4, 86, 1, 4
5, 78, 1, 88, 27, 174, 31, 57, 24, 53, 74, 5, 12, 5, 69, 6, 83, 78, 58, 18
, 27, 14, 77, 54, 4, 80, 39, 124, 7, 57, 20, 71, 3, 79, 112, 145, 82, 25,
27, 4, 21, 1, 75, 17, 68, 2, 3, 85, 34, 41, 70, 28, 84, 80, 2, 3, 45, 61,
49, 40, 38, 68, 75, 25, 80, 155, 79, 87, 1, 65, 89, 238, 78, 82, 36, 39, 4
, 3, 2, 90, 5, 57, 7, 143, 25]
Total batches: 311
Total products: 9661.
```

```
Batch List After:[72, 76, 71, 75, 82, 75, 84, 81, 73, 76, 80, 76, 78, 79,
74, 83, 79, 75, 79, 78, 79, 74, 78, 73, 77, 76, 77, 76, 72, 77, 80, 80, 76
, 73, 85, 76, 81, 78, 71, 77, 81, 80, 78, 80, 74, 72, 73, 77, 80, 79, 75,
74, 78, 68, 80, 81, 76, 77, 75, 78, 74, 75, 72, 77, 79, 78, 79, 79, 74, 79
, 74, 80, 79, 80, 77, 78, 70, 76, 73, 77, 76, 76, 76, 79, 83, 77, 77, 75,
78, 75, 77, 81, 76, 82, 77, 79, 70, 76, 82, 78, 74, 74, 78, 77, 82, 73, 76
, 79, 70, 80, 75, 73, 77, 74, 75, 79, 74, 77, 79, 79, 79, 77, 77, 74, 76,
75, 77, 72, 77, 75, 14]
Total batches: 131
Total parts produced: 9996.
```