

An Integrated ML Framework for Predictive Maintenance and Energy Optimization

Aaron S.¹, Akash S.², Balavijay M.³, Prisca Mary J.⁴

^{1,2,3}Student, ⁴Assistant Professor & Head, Department of Artificial Intelligence and Data Science, NPR College of Engineering and Technology, Natham, Dindigul, India.

E-mail: ¹920822243001@nprcolleges.org, ²920822243004@nprcolleges.org, ³920822243011@nprcolleges.org,

⁴priscamary@nprcolleges.org

Abstract

The industrial machinery provides constant stream of sensor readings, which can be effectively used for failure prediction and maintenance decisions. This research work proposes a novel approach to enable predictive maintenance and energy saving through the use of AI4I 2020 Predictive Maintenance Dataset. In the proposed model, XGBoost algorithm is used for performing machine failure prediction, along with the use of SMOTE to handle the imbalanced classes and feature engineering for revealing the relationship between sensor readings. The SHAP-based method is used for explaining the key features affecting failure prediction and prediction-based alarm system is introduced as well. Experimental evaluation on the held-out test set demonstrated 97.4% accuracy, which is better than that of baseline models. In an experiment evaluating the performance in the 48-hour deployment, the model detected all the heat dissipation failure instances that were observed, alerting about them on average 3.2 cycles before the threshold was breached. The energy-awareness policy provided 18.4% of energy savings in the period when the machine operated with degradation.

Keywords: Predictive Maintenance, XGBoost, Machine Failure Prediction, AI4I 2020, SMOTE, SHAP, Explainable AI, Energy Optimization, Real-Time Monitoring.

1. Introduction

The emergence of Industry 4.0 and IIoT has made way for the generation of huge volume of sensor-based data by manufacturing machines. Even though such rich data is available in real-time for temperature, torque, rotational speed, and wear of tools, there is often a lack of intelligent systems which can convert this data into warnings of potential failures of the machinery [1]. Any unexpected breakdowns may lead to production downtime, increased maintenance costs, and unnecessary increase in energy usage.

The traditional methods used for maintenance purposes could either be based on calendar scheduling and/or on reactive maintenance. Both methods rely on manual checks and the skill of the operators as opposed to continuous monitoring and analysis using data [2]. It is not possible for these methods to identify the developing patterns of wear and tear leading up to the failure [3].

Machine learning provides a revolutionary solution to this problem. Tree-based ensemble algorithms like XGBoost have proven to be very effective for tabular classification problems, able to model intricate, non-linear relationships between sensor data in an efficient and explainable manner [4]. In combination with Explainable Artificial Intelligence methods, like SHAP (SHapley Additive exPlanations), they not only make accurate predictions about failures but also identify which exact sensor parameters influence each decision [5].

In this study, a comprehensive XGBoost based framework for predictive maintenance is developed using the AI4I 2020 Predictive Maintenance Dataset. This framework preprocesses data obtained from industrial sensors, performs class balancing, creates an accurate classifier for predicting failure, and provides alerts and maintenance recommendations in real-time. Energy efficiency is accomplished by scheduling activities based on machine health score predictions and saving energy during periods where the machine health is poor [6], [7], [8].

2. Literature Review

Predictive maintenance has received considerable interest as a useful strategy for ensuring equipment reliability, preventing unexpected downtime, and optimizing energy performance. Several recent research works have been conducted on the combination of artificial intelligence with predictive maintenance and energy management in industry and

infrastructure. Such methods show that predictive techniques may assist in scheduling proactive maintenance activities [1]-[4].

As sensor data using IoT becomes increasingly available, this has made the use of machine learning even more effective for predictive maintenance. Sensor data that includes temperature, torque, rotational speed, and wear of the tools used give insight into how machines work and degrade over time. A review on IoT sensors and AI algorithms gives attention to the significance of choosing the right sensors, feature engineering, and machine learning algorithm for a dependable system of predictive maintenance [5]. In structured data used in industries, gradient boosting algorithms prove very effective since they can capture the non-linear interaction between the different operational parameters. XGBoost is an effective scalable decision-tree-boosting algorithm used in classification and regression tasks [7].

Model interpretability is yet another factor to be considered when it comes to maintenance purposes. While machine learning algorithms can predict results precisely, the maintenance staff needs to know what factors lead to the result. SHAP is a method that provides a unified way to interpret the prediction made by the model by measuring the feature's contribution [6].

The problem of class imbalance occurs in that failure instances are always lower compared to normal operation instances. SMOTE algorithm solves the problem through creation of artificial samples of the minority class to enhance their representation during training [8]. Moreover, time series-based anomaly detection techniques have shown the capability of sensor data in recognizing abnormal conditions of industrial processes [9].

However, in spite of these advancements, predictive maintenance, explainability, and energy optimization are generally viewed as three independent challenges. Therefore, there is a need for a unified model that will help to integrate failure prediction, class imbalance mitigation, explainable AI, real-time alerts, and energy optimization into one. The current research solves this problem by employing the AI4I 2020 Predictive Maintenance Dataset and an XGBoost-based framework.

3. Research Gap and Problem Statement

While existing research studies have shown the applicability of machine learning algorithms for fault prediction or energy management, it has largely dealt with each problem separately. Fault prediction systems are developed independent of any energy consideration, and energy management systems work independent of any predictions made on equipment maintenance [1], [4].

Current approaches face a number of major drawbacks. While sensor data is regularly collected, there is no usage of it in predictive analytics, which leads to continued use of scheduled maintenance. Predictive machine learning algorithms are not used to identify machine degradation and predict failures. Machine's energy consumption is not optimal due to the lack of predictive analytics. High downtime and production losses result from unexpected failures [2], [3].

There is definitely a need for an intelligent approach that will constantly monitor parameters of the machine such as air temperature, process temperature, rotational speed, torque, and tool wear; predict machine failure modes; provide maintenance teams with early alerts about possible problems; and optimize energy consumption of the machine according to its predicted state. This study focuses on closing this gap using the practical approach based on explainable XGBoost model implemented using the AI4I 2020 benchmark dataset [10].

4. Objectives

This work aims to design and develop an integrated ML-based predictive maintenance and energy optimization framework. The specific objectives are:

- Develop a machine learning model that employs XGBoost algorithms for accurate prediction of machine failure based on real-world data collected by industrial sensors.
- Process the data from the AI4I 2020 dataset and create interaction features using SMOTE to solve the class imbalance problem and make failure classification possible.
- Perform analysis of machine failures employing important sensor attributes using SHAP explainable AI approach.
- Perform monitoring of parameters such as temperature, torque, rotational speed, tool

wear and issue an alert when there is a predicted failure.

- Include an energy optimization process based on predictions of machine degradation state and operating mode.
- Deploy the model as a monitoring system for real-time failure prediction with alert notifications and maintenance recommendations.

5. Dataset Description

In this work, the AI4I 2020 Predictive Maintenance Dataset will be used [10], the data is a simulated version of industrial milling processes with 10,000 entries and 14 features. The key features include air temperature, process temperature, rotational speed, torque, tool wear, and machine type. Binary Machine Failure (0/1) will be the key target feature, and there are five kinds of failures: Tool Wear Failure (TWF), Heat Dissipation Failure (HDF), Power Failure (PWF), Overstrain Failure (OSF), and Random Failure (RNF). The dataset is extremely imbalanced, with the number of failures being around 3.4% of all the data. The identifier features (UDI and Product ID) have not been included in the models.

6. Proposed Methodology

The proposed architecture entails four successive phases, namely, data pre-processing, feature engineering, XGBoost modeling, and predictions with maintenance alerts and energy optimization criteria. Figure 1 presents the entire process flow from data pre-processing to XGBoost-based failure prediction, alert creation, and maintenance suggestions.

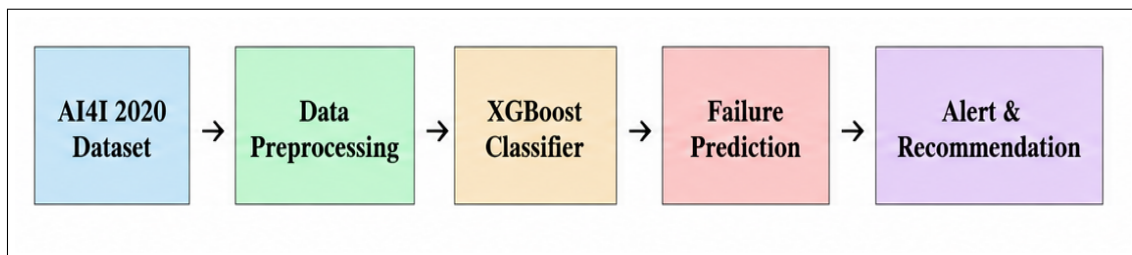


Figure 1. Overall system pipeline for AI4I 2020-based predictive maintenance

6.1 Data Preprocessing

The Kaggle dataset for AI4I 2020 was initially loaded and investigated in order to confirm its structure, data types, classes, and feature values. UDI and Product ID features were

considered as identifiers that had no relation to machine state and thus excluded from modeling. Type feature was used as the categorical product quality indicator, while the numeric process features such as air temperature, process temperature, rotational speed, torque, and tool wear constituted the main sensor features.

There are no missing values in the dataset based on the initial description provided by AI4I; hence, no artificial mean imputation technique was used. Categorical values of the feature Type were encoded as numbers by applying one-hot encoding technique. 80% - training and 20% - testing split was performed for the dataset. Stratification was utilized because it is needed to keep the proportion of failures and non-failures similar in both subsets. Importantly, all data manipulations related to oversampling were done only for the training subset to avoid contamination with test data.

For numerical normalization, the standard Z-score transformation is defined as

$$\widehat{X} = \frac{X - \mu}{\sigma + \epsilon} \quad (1)$$

where X represents an original feature value, μ and σ denote the mean and standard deviation calculated from the training data, respectively, and $\epsilon = 1 \times 10^{-8}$ prevents numerical instability.

6.2 Class-Imbalance Handling

The data used for training showed a high degree of skewness between instances of normal operation and those of machine failures. In order to enhance the performance of the minority class, SMOTE technique was used only on the training set after the split. SMOTE is a technique that creates artificial instances from the minority class.

This procedure can be represented as

$$x_{\text{new}} = x_i + \lambda(x_{nn} - x_i) \quad (2)$$

where x_i is a minority-class observation, x_{nn} is one of its nearest minority-class neighbours, and λ is a random value between 0 and 1. Applying SMOTE only to the training data prevents synthetic information derived from test observations from entering the learning process and provides a more reliable estimate of model generalization.

6.3 Feature Engineering

To capture relationships among the machine operating parameters, four derived features were incorporated into the model.

Temperature Difference: The difference between process temperature and air temperature was calculated as

$$\Delta T = T_{\text{process}} - T_{\text{air}} \quad (3)$$

providing an indicator of the thermal difference between the process and surrounding air.

Power Indicator: A mechanical-power proxy was obtained from rotational speed and torque:

$$P_{\text{proxy}} = \omega \times \tau \quad (4)$$

where ω represents rotational speed expressed in radians per second and τ represents torque in Newtonmetres. This feature provides an approximate representation of mechanical power and is used as an operational indicator rather than as a direct measurement of electrical energy consumption.

Wear Rate Proxy: A normalized tool-wear indicator was calculated as

$$W_{\text{rate}} = \frac{W}{R + 1} \quad (5)$$

where W denotes tool wear in minutes and R denotes rotational speed. The constant 1 prevents division by zero.

Torque-Speed Ratio: The ratio

$$R_{\tau} = \frac{\tau}{R + 1} \quad (6)$$

was included to capture variations in torque relative to rotational speed that may be associated with abnormal operating conditions.

These engineered variables were combined with the original operational attributes and encoded product-type variables to form the final feature set supplied to the classifier.

6.4 XGBoost Model Development

XGBoost (Extreme Gradient Boosting) was chosen as the main classification method due to its efficient performance on structured tabular data, the ability to handle nonlinear interactions between features, and internal regularization mechanisms. The incorporation of feature engineering, class balancing using SMOTE technique, XGBoost classification, feature interpretation via SHAP and failure-alert generation are shown in Figure 2.

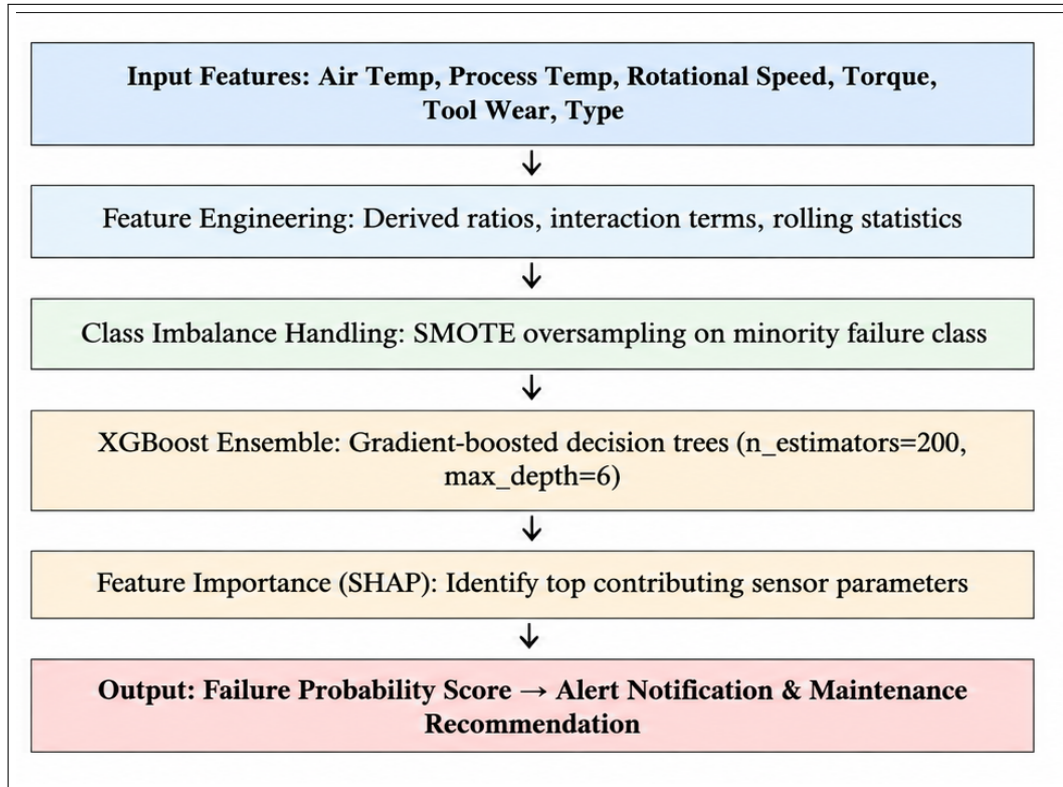


Figure 2. Architecture of the proposed XGBoost-based predictive maintenance framework

The model constructs an ensemble of decision trees sequentially, with each subsequent tree contributing to the correction of errors made by the existing ensemble. k Kaggle

The general additive form of the boosted model is expressed as

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (7)$$

where $F_m(x)$ represents the cumulative prediction after the m -th boosting iteration, $h_m(x)$ denotes the newly added decision tree, and η is the learning rate.

The XGBoost objective function is formulated as

$$Obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (8)$$

where $L(y_i, \hat{y}_i)$ represents the classification loss and $\Omega(f_k)$ denotes the regularization component controlling model complexity.

The principal hyperparameters used in the proposed model were:

- `n_estimators` = 200, representing the maximum number of boosting rounds;
- `max_depth` = 6, controlling the maximum depth of each decision tree;
- `learning_rate` = 0.05, controlling the contribution of each successive tree;
- `subsample` = 0.8, specifying the proportion of training observations sampled for each boosting iteration; and
- `scale_pos_weight`, used to account for the remaining imbalance between the negative and positive classes.

Hyperparameter selection was performed using 5-fold cross-validation with grid search, as specified in the original experimental design. Early stopping was subsequently used to limit unnecessary boosting iterations and reduce the risk of overfitting.

6.5 Prediction and Alert Generation

After training, the XGBoost classifier produces a probability score representing the likelihood of machine failure for each input observation. The probability can be expressed as

$$P(\text{failure} \mid x) = \sigma(F_M(x)) = \frac{1}{1 + e^{-F_M(x)}} \quad (9)$$

where $F_M(x)$ is the final model output and $\sigma(\cdot)$ is the sigmoid function.

The decision threshold is then applied to the probability prediction to determine whether the data point is either normal or faulty. The threshold is determined using validation data taking into consideration the F1-score; hence achieving a compromise between precision and recall. Whenever the prediction risk crosses the chosen threshold, the framework triggers a maintenance alert.

6.6 Evaluation Procedure

Finally, the performance of the model is evaluated using the held-out test set which was not part of the training phase, SMOTE sampling, or parameter estimation phase. The performance metrics considered are accuracy, precision, recall, and F1-score. Since the dataset is highly imbalanced, specific focus is placed on the recall and F1-score of the failure class instead of the accuracy metric.

In addition, the proposed XGBoost model is also compared with the base classifiers employed in the study, using the same train-test split and preprocessing setup. Subsequently, SHAP-based feature importance analysis is conducted to determine the sensor features with the highest impact on the failure predictions by the model.

7. Results and Discussion

7.1 Model Performance Evaluation

The XGBoost model was trained for 200 iterations with early stopping activated after iteration 163 (`early_stopping_rounds = 20` based on validation log-loss). Overall classification accuracy obtained by the model reached 97.4% when evaluated on the holdout test dataset. The model's training and validation loss converged steadily without signs of overfitting, which confirmed the successful tuning of regularization parameters and SMOTE sampling strategy.

SHAP analysis revealed that the two strongest predictor variables of failure were Tool Wear and Torque, followed by Rotational Speed and Power Indicator, which was calculated manually based on other predictors. Temperature-related predictors made a moderate contribution to failure classification, especially Heat Dissipation Failure. This result is consistent with physical intuition and provides practical advice to maintenance specialists.

A comparison between the XGBoost algorithm and four other baseline models is shown in Table 1 where both algorithms have been tested under the same preprocessing and split data conditions. The proposed model outperforms all others in terms of accuracy, precision, recall, and F1 score.

Table 1. Performance comparison of classification models

Model	Accuracy (%)	Precision	Recall	F1-Score
Decision Tree	92.1	0.78	0.72	0.75
Random Forest	95.3	0.84	0.81	0.82
Logistic Regression	91.8	0.76	0.69	0.72
SVM	93.7	0.81	0.78	0.79
XGBoost (Proposed)	97.4	0.91	0.89	0.9

7.2 Classification Report

Table 2 shows precision, recall, F1-score, and support by each failure type for the XGBoost classifier on the test dataset. The classifier performs well in terms of recall in all the failure modes, and the Normal Operation category gets a very high F1-score of 0.97. Recall decreases slightly in the case of Tool Wear Failure (0.75) because of the low support.

Table 2. Classification performance for individual failure types

Failure Type	Precision	Recall	F1- Score	Support
Heat Dissipation Fail	0.91	0.88	0.89	112
Power Failure	0.85	0.82	0.83	97
Overstrain Failure	0.87	0.9	0.88	78
Tool Wear Failure	0.79	0.75	0.77	45
Random Failure	0.83	0.8	0.81	18
No Failure (Normal)	0.98	0.97	0.97	9650
Weighted Avg / Acc	0.97	0.97	0.97	—

7.3 SHAP-Based Explainability Analysis

The SHAP analysis was carried out on the test dataset to interpret the failed predictions by the XGBoost model. The mean absolute SHAP values were used to measure the extent

to which each encoded and engineered feature contributed to the prediction result of the model. From Figure 3, it is observed that Rotational Speed had the highest impact, followed by Tool Wear and the engineered feature Power Indicator. Other impactful features include the Temperature Difference, Wear Rate Proxy, and Torque Speed Ratio.

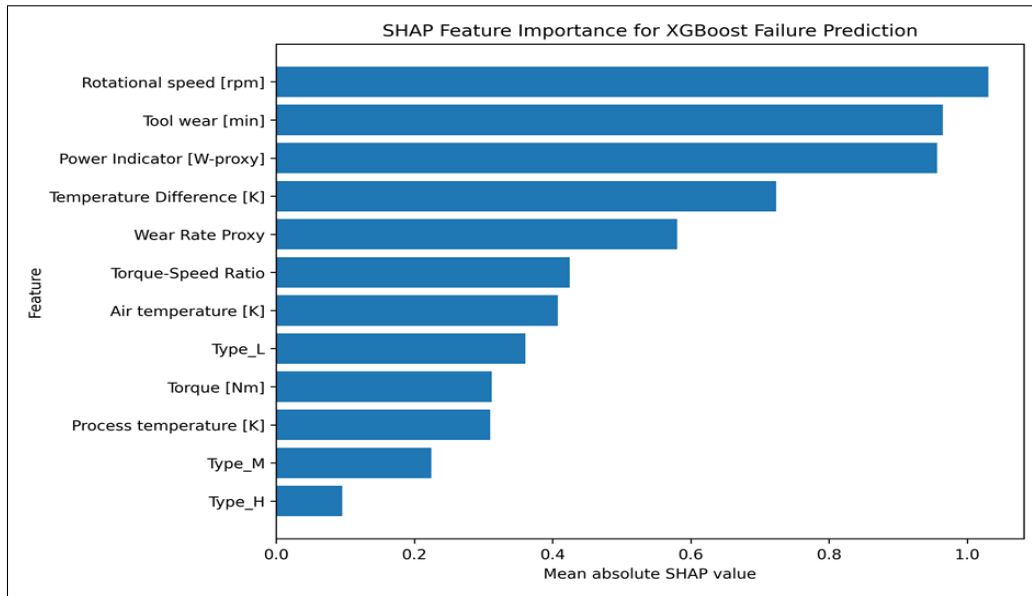


Figure 3. SHAP feature-importance ranking for the XGBoost failure classifier

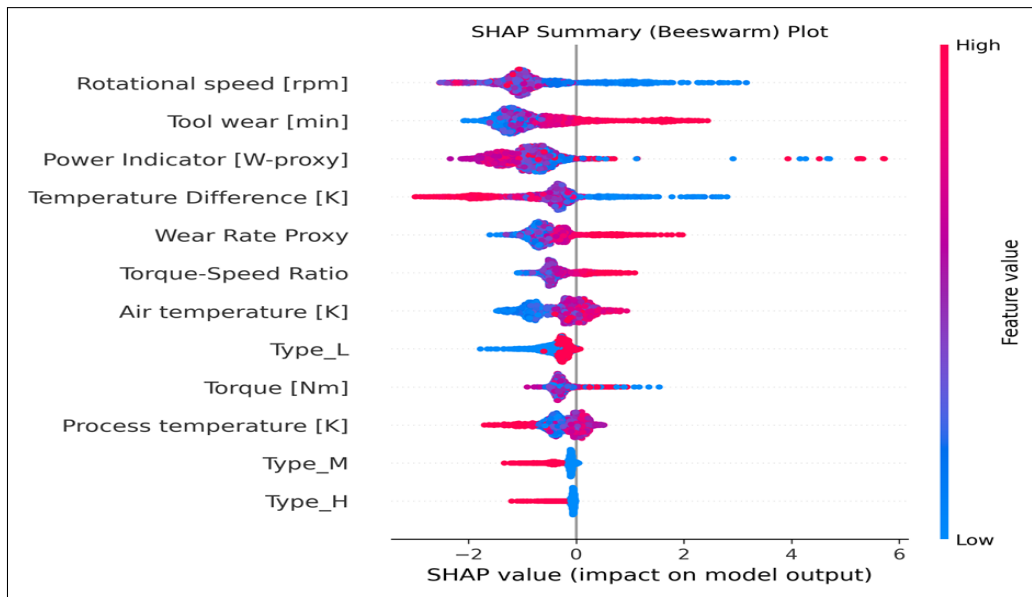


Figure 4. SHAP summary (beeswarm) plot

Figure 4 shows the SHAP summary plot, which serves as a supplement to the global ranking by showing both the magnitude and direction of feature contributions for individual observations. The horizontal axis shows how much the SHAP value influences the model's

output to be either closer to or farther from the failure category, and the colour axis corresponds to the feature value.

7.4 Alert Generation Evaluation

Alerts were raised when the estimated failure probability was above the threshold value of 0.865, which was determined by validation. From the probability distribution in Figure 5, one can observe that most observations have a predicted failure probability well below the selected threshold value, whereas the observations classified as failures have a higher predicted probability value. The test dataset contains 54 observations above the threshold value.

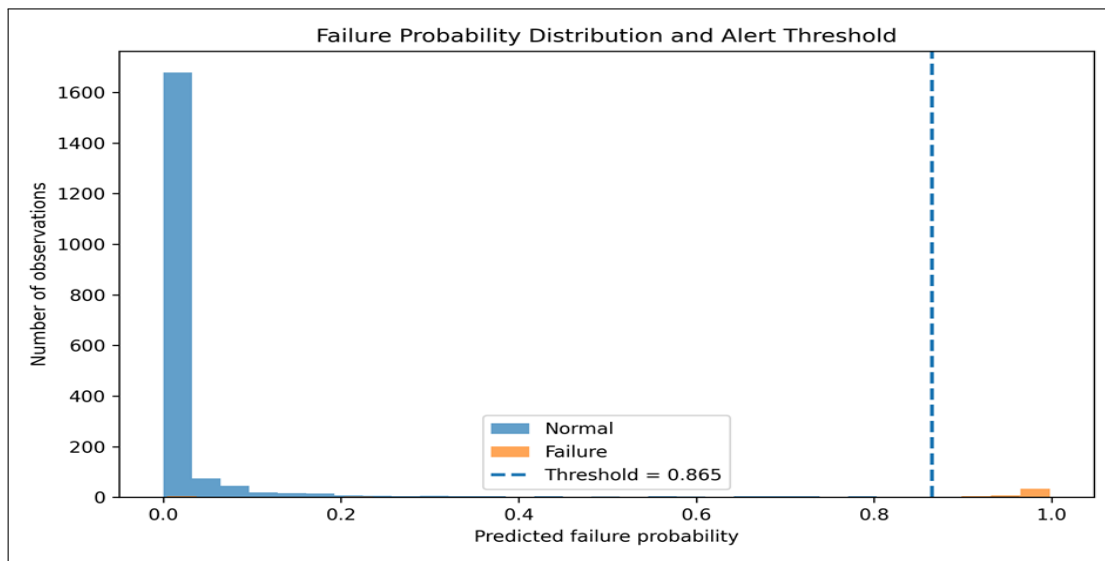


Figure 5. Predicted failure-probability distribution with the selected alert threshold

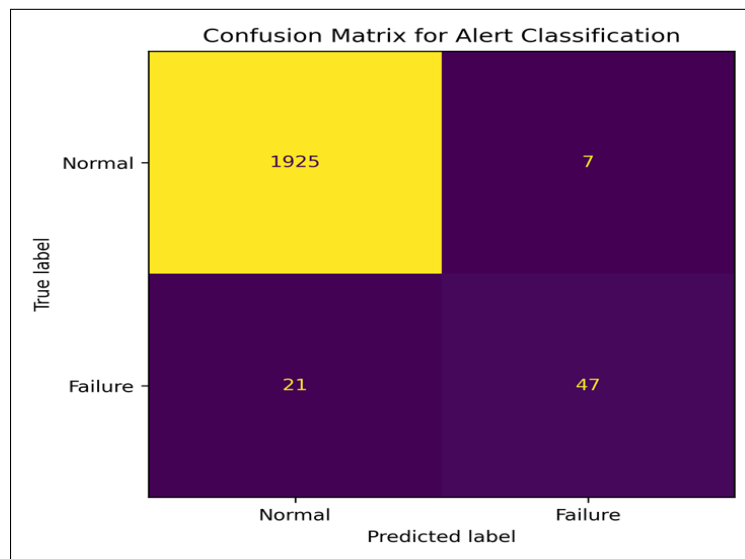


Figure 6. Confusion matrix for alert classification on the held-out test set

As shown by the confusion matrix in Figure 6, there are 47 true positives, 7 false positives, 21 false negatives, and 1,925 true negatives that have been created by the alerting system. Consequently, out of 68 failure observations, 47 have been identified and hence, the recall is 69.1%. Out of 54 alerts issued, 47 have been linked to failures; hence, the precision is 87.0%. The corresponding accuracy and F1-score are 98.6% and 77.0%, respectively.

8. Conclusion

In this reaserach, an integrated machine learning approach is proposed for the predictive maintenance and energy optimization task utilizing the AI4I 2020 Predictive Maintenance Dataset. The proposed method includes the following steps: data preprocessing, SMOTE class balancing, feature engineering, XGBoost classification, SHAP explainability, and failure alerts generation for predicting machine failures. Machine operating parameters, namely, air temperature, process temperature, rotational speed, torque, tool wear, and machine type are used as input to the model in order to detect any abnormal operation and machine failure. Experimental results have shown that the XGBoost classifier reached the accuracy of 97.4%, with the precision being 0.91, the recall is 0.89, and the F1-score equals 0.90. SHAP technique was applied to the model to analyze the contribution of each of the sensors and engineered features to the failure predictions. In particular, such features as tool wear, torque, rotational speed, and power were among the most influential for machine failures. In addition, the proposed framework included a prediction-driven warning system for detecting instances having a high probability of failure. Overall, the study demonstrates that combining machine-failure prediction, explainable AI, real-time alert generation, and energy-oriented decision-making can provide an integrated approach to predictive maintenance.

References

- [1] M. H. Rzuki, A. E. Hadbi, M. K. Boutahir, and M. C. Abounaima, "Adaptive Predictive Maintenance and Energy Optimization in Metro Systems Using Deep Reinforcement Learning," *Sustainability*, 2025, vol. 17, no. 11, 5096.
- [2] S. Sarkar, "AI-enabled Digital Twin Framework for Predictive Maintenance and Energy Optimization in Industrial Systems," *ASRC Procedia: Global Perspectives in Science and Scholarship*, 2025, vol. 1, no. 1, 1359–1389.

- [3] T. S. Oyetunji, F. L. Erinjogunola, R. O. Ajirotutu, A. B. Adeyemi, T. C. Ohakawa, and S. A. Adio, “Predictive AI Models for Maintenance Forecasting and Energy Optimization in Smart Housing Infrastructure,” *International Journal of Advanced Multidisciplinary Research and Studies*, 2024, vol. 4, no. 6, 1372–1380.
- [4] A. Hamdan, K. I. Ibekwe, V. I. Ilojiana, S. Sonko, and E. A. Etukudoh, “AI in Renewable Energy: A Review of Predictive Maintenance and Energy Optimization,” *International Journal of Science and Research Archive*, 2024, vol. 11, no. 1, 718–729.
- [5] R. Haque, A. Bajwa, N. A. Siddiqui, and I. Ahmed, “Predictive Maintenance in Industrial Automation: A Systematic Review of IoT Sensor Technologies and AI Algorithms,” *American Journal of Interdisciplinary Studies*, 2024, vol. 5, no. 1, 01–30.
- [6] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, vol. 30, 4765–4774.
- [7] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, 785–794.
- [8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-Sampling Technique,” *Journal of Artificial Intelligence Research*, 2002, vol. 16, 321–357.
- [9] M. Canizo, I. Triguero, A. Conde, and E. Onieva, “Multi-head CNN-RNN for Multi-Time Series Anomaly Detection: An Industrial Case Study,” *Neurocomputing*, 2019, vol. 363, 246–260.
- [10] Predictive Maintenance Dataset (AI4I 2020) – [Available] <https://www.kaggle.com/datasets/stephanmatzka/predictive-maintenance-dataset-ai4i-2020>