Research on the Efficiency Improvement Model of Lighthouse Factory Operation Based on XGBoost Algorithm

Wei Song
School of Digital Economics, Changzhou College of Information Technology, Changzhou 213164, Jiangshu, China
Corresponding author: Wei Song (e-mail: 13051039320@163.com)

ABSTRACT This study proposes an operational efficiency improvement model for Lighthouse Factories based on the XGBoost algorithm. By integrating large-scale heterogeneous data from PLCs, MES, SCADA systems, maintenance records, and energy monitoring platforms, the model identifies operational inefficiencies and predicts critical production events through advanced feature engineering and cost-sensitive learning strategies. A comprehensive data-to-decision framework is established, enabling predictive maintenance, dynamic scheduling, and near real-time decision support. Experimental evaluation on an 18-month industrial dataset containing over 15 million records demonstrates superior performance compared with conventional machine learning approaches, achieving higher predictive accuracy, reduced unplanned downtime, and improved production throughput. The proposed framework is particularly applicable to Industry 4.0 environments supported by industrial wireless communication networks, antenna-enabled Industrial Internet of Things (IIoT) infrastructures, and cyber-physical manufacturing systems, where reliable data acquisition and transmission are essential for intelligent operation. The results confirm the effectiveness, scalability, and practical value of the proposed model for enhancing operational efficiency in smart manufacturing environments.

INDEX TERMS xgboost algorithm, lighthouse factory, predictive maintenance, industrial wireless communication

I. INTRODUCTION

A. BACKGROUND INFORMATION

A Lighthouse Factory, a concept popularized by the World Economic Forum (WEF), is a manufacturing facility recognized as a world leader in implementing and adopting advanced Industry 4.0 technologies (e.g., IoT, AI, cloud computing) across its value chain, enabling end-to-end data flow and real-time decision-making through advanced analytics. These factories serve as "beacons" of innovation, demonstrating how production processes can be reinvented through the integration of cyber-physical systems (CPS), the Internet of Things (IoT), and digital twins. In the textile industry, for example, a Lighthouse Factory leverages these technologies to create fully automated, end-to-end digital production lines, from customized fiber spinning to final garment cuttng, showcasing the future of efficient and sustainable manufacturing [1]. These factories use real-time data capture, strong automation, and machine-to-machine communication to permit versatile, productive, and resilient production. A typical lighthouse factory incorporates intelligent robotics, analytics platforms in the cloud, and predictive maintenance tools, making human operators and machine functions coexist harmoniously.

In lighthouse factories, operational efficiency is a significant performance metric that directly affects throughput, overall equipment effectiveness (OEE), energy, product quality, and production cost [2]. Because such factories are frequently used as examples of industrial best practices, maximizing efficiency will pay dividends in more ways than one, since other manufacturers will find it easier to follow the example by improving their productivity. Due to high operational efficiency, there is minimal downtime, resource allocation, and increased adaptation to the market requirements. Given the imperatives of sustainability objectives and the fierce environment of global competition, the challenge of accomplishing more output with fewer resources, all while maintaining high quality, is necessary not only for economic viability but also for achieving environmental sustainability. This drive for radical efficiency is especially acute in the textile industry, where minimizing water and energy consumption in resource-intensive processes like dyeing and finishing, or reducing material waste through precise cuttng, is vital for both profitability and meeting increasingly strict global environmental standards.

B. MOTIVATION & GAP ANALYSIS

Although technologically advanced, lighthouse factories encounter many difficulties concerning efficiency, maintenance, and improvement. The data generated in such environments are typically complex multi-source time-series streams, originating from heterogeneous sensors and industrial systems [3]. Such records are nonlinear, have different space and time resolutions, and are sometimes inconsistent. Equipment failures or inefficiency events are often rare but highly impactful, making their prediction a challenging class imbalance problem [4]. Also, operational choices often must be taken in real time, so its models must be both precise and computationally inexpensive.

Classical statistical models, while interpretable, often struggle with capturing the nonlinear, high-dimensional interactions present in manufacturing data [5]. Conversely, deep learning approaches, particularly in their "black-box" form, can model complex relationships but typically require vast amounts of labelled data, significant computational resources, and offer limited interpretability—an important drawback in industrial environments where decision accountability is essential [6]. Moreover, deep neural networks can be complex to deploy on edge devices or integrate into existing manufacturing execution systems (MES) due to their resource demands and opaque decision-making processes.

The Extreme Gradient Boosting (XGBoost) algorithm offers a balanced solution to these challenges. As a treebased gradient boosting framework, XGBoost is well-suited for heterogeneous tabular data, can natively handle missing values, and efficiently model nonlinear feature interactions [7]. It also learns comparatively fast compared to numerous deep learning systems, and its output can be interpreted into an uncommon operator trusting ranking of features. Moreover, the scalability of XGBoost and its support of custom loss functions make the later especially suitable for manufacturing applications, such as cost-sensitive tasks, in which false negatives have a disproportionately large effect.

C. RESEARCH CONTRIBUTIONS AND PAPER STRUCTURE

Some contributions to the area of intelligent manufacturing analytics are evident in this paper: (a) Development of a complete data-to-decision pipeline for operational efficiency improvement in lighthouse factories using XGBoost as the core predictive model. (b) Introduction of feature engineering methods tailored to manufacturing data, including temporal aggregations, rolling statistical measures, and machine interaction features that capture dependencies across equipment. (c) Integrating hyperparameter tuning and operational constraints into the modelling process, focusing on cost-sensitive loss functions to prioritize highimpact predictions. (d) Extensive experimental evaluations against multiple baseline models, supplemented by a real-world deployment simulation, are performed to assess practical viability [8].

These contributions address a critical research gap where prior work has often focused on isolated components such as predictive maintenance or quality prediction without presenting a unified, operationally deployable pipeline [9]. The proposed framework will achieve performance optimization in predictive accuracies and can be interpreted, scaled, and work within the context of industrial systems already in use. Through rigorous feature engineering, algorithmic optimization, and real-world integration, this study offers a replicable methodology for improving manufacturing efficiency at scale [10].

The rest of this paper is organized as follows: Section "LITERATURE REVIEW" is a literature review related to efficiency enhancement in smart factories, machine learning usage in manufacturing, and XGBoost implementation in the industrial settng. In section PROBLEM STATEMENT & OBJECTIVES, the problem is stated formally, and the objectives and constraints are indicated. Section METHODOLOGY is the methodology section that presents the system architecture, the data collection process, the feature engineering, the model design, and the training. In section EXPERIMENTS & RESULT, the experiment and its setup are reported along with its results and ablation analyses, and then a simulation of an actual deployment is run. Section DISCUSSION talks about the interpretation of the results, practical implications, and limitations. Section POLICY IMPLICATIONS & RECOMMENDATIONS provides policy recommendations, which lead to Section CONCLUSION & FUTURE WORK, which concludes by giving a summarised conclusion and suggesting future research.

II. LITERATURE REVIEW

A. EFFICIENCY IMPROVEMENT IN SMART FACTORIES

Efficiency improvement in smart factories is a long-standing focus of manufacturing research, often tackled through operations research (OR) methodologies such as scheduling optimization, resource allocation, and maintenance planning. The usual techniques applied to streamline throughput and minimize costs are such as linear programming, mixed-integer programming, and heuristic algorithms [11]. The techniques enable the manufacturers to streamline the process through the ease of job scheduling and machine availability so that the manufacturers establish optimal performances.

Overall Equipment Effectiveness (OEE) is a baseline performance indicator to measure manufacturing performance through the combination of availability levels, performance efficiency, and quality rates [12]. It has also been revealed through studies that incremental gains in OEE, even by a small amount of 5-10% are capable of generating impressive volumes of production, in some occasions amounting to 8% [13]. All these benefits can be summarized into terms of savings on costs and greater sustenance of resources, hence signifying the significance of efficiency of operations in the modern-day factory economically and environmentally.

Predictive maintenance has evolved from traditional statistical process control (SPC) and reliability-centred maintenance (RCM) models, including failure distribution analysis such as Weibull and exponential models. Recent hybrid OR-ML frameworks combine predictive maintenance scheduling with dynamic production planning, reducing downtime by anticipating failures without disrupting workflows [14]. Despite these achievements, most of the systems will approach maintenance and scheduling as two separate entities, neglecting the possibilities of comprehensive optimization that would consider various operational KPIs and optimize them altogether.

B. MACHINE LEARNING IN MANUFACTURING

Machine learning has become a popular tool as an effective mechanism for addressing very intricate issues in manufacturing, such as failure prediction, quality inspection, and energy optimization. Supervised learning models such as logistic regression, decision trees, support vector machines, and neural networks have been applied successfully to classify faults and predict machinery breakdowns [15]. Research findings indicate that including ML-based predictive maintenance can cut downtime by up to 50 percent and considerably increase operational availability.

For quality control, convolutional neural networks (CNNs) have revolutionized visual defect detection by automating inspection with high accuracy [16]. Meanwhile, tree-based models such as random forests and gradient boosting are favored for sensor-based classification of product quality due to their robustness to noisy data [17]. On the energy front, reinforcement learning and regression models have optimized consumption paterns, achieving energy savings between 8% and 15% without compromising output [18]. However, numerous ML applications are still localized and focused on one goal instead of efficiency in the entire factory.

Another important issue is the data heterogeneity and volume because manufacturing environments produce different data streams of PLCs (Programmable Logic Controller), MES (Manufacturing Execution System), SCADA (Supervisory Control and Data Acquisition), and environmental sensors. Their samples are inconsistent and have gaps. Though data fusion techniques have been suggested to reconcile such multi-source data, litle research has been made in conjunction with domain-specific feature engineering and cost-sensitive modelling to deal with real-time decisions. Lacking these, ML solutions face the threat of low generalizability and workplace applicability in industry because they pose interpretability and deployment issues.

C. XGBOOST AND TREE-BASED METHODS IN INDUSTRY

Tree-based ensemble models, including Random Forest, Gradient Boosting Machines (GBM), and XGBoost, are widely regarded as state-of-the-art for tabular data in industrial applications [19]. XGBoost, especially, brings together the concepts of scalability, accuracy, and flexibility through efficient processing of missing data and modelling of complex feature interactions. It finds applications in predicting tool wear, fault detection, and adaptive production scheduling. Such victories indicate that it is feasible in the manufacturing environment in which data volume and velocity are high.

XGBoost, in contrast to deep learning models, provides increased interpretability with the assistance of feature importance metrics and the capability of combining with the explainability software, such as SHAP values, to atribute remarkably equal scores to features within each estimation. Such visibility contributes to the trust of the operators and informs corrective action on the factory processes. Furthermore, a relatively low computational complexity allows XGBoost to be deployed to edge devices and used in real-time inference, overcoming latency limits of importance in production decision-making.

D. GAPS IN PRIOR WORK

Despite progress, the literature reveals gaps in integrating multi-source factory data into a single, operational pipeline that combines domain-aware feature engineering, cost-sensitive objective functions, and interpretable gradient boosting models within lighthouse factories [20]. In most studies, the researchers concentrate specifically on individual tasks such as predictive maintenance or predicting the quality of the system rather than integrating the complete system or considering implementation in real-time. Additionally, limited research offers extensive ablation and sensitivity analyses to evaluate model robustness under noisy or missing data conditions—common in industrial settngs—and rarely addresses seamless integration into existing MES/SCADA systems for real-world use [21]. The proposed research work will improve upon these gaps by suggesting a universally applicable XGBoost-based framework that balances predictive performance, model interpretability, and operation limitations to enhance scale in smart factories with effectiveness.

III. PROBLEM STATEMENT & OBJECTIVES

The research will establish a predictive-prescriptive system that predicts the occurrence of an operational inefficiency phenomenon, e.g., equipment failing, process botlenecks, or quality issues in a lighthouse factory environment. The system's main aim will be to decrease the unplanned downtime by a minimum of 10 percent per week, grow throughput by 5 percent, and improve the first-pass quality yield of 3 percent or above, working with data-driven interventions using multi-source sensor and operational data.

The system must produce the predictions within 500 milliseconds, owing to operational limitations that should facilitate real-time decision-making. It should give explanatory sub-products so human operators can comprehend and trust model suggestions. On-premises deployment is required due to data privacy limitations, and sensitive manufacturing

data should be stored during on-premises deployment. The system should also be flexible and fit process changes without retraining. Finally, the purpose is to prove that even a more traditional pipeline, constructed with XGBoost, complemented with a custom feature engineering process and cost-sensitive learning targets, may be more effective than more basic techniques and capable of addressing real-world deployment, thus making a step forward in intelligent manufacturing analytics and decision support.

IV. METHODOLOGY

A. OVERALL SYSTEM ARCHITECTURE

The architecture of the proposed operation efficiency improvement system is adopted to provide a smooth combination of data collection, data processing, predictive modelling, and resultant decision-making processes within the environment of a lighthouse factory. The system is initiated by implementing different industrial sensors in production lines that constantly monitor the state of machines and manufacturing conditions and processes. These sensors use the raw data in an edge computing layer where preliminary processing is done to minimise noise and delays of transmissions, which are crucial in real-time reaction to critical scenarios.

Pre-processed data is moved to a centralized data lake, a scaled repository to store structured and unstructured data from varying sources. This would have a designated feature store to extract and save the machine learning features engineered at this repository. This feature set is then run as the core predictive model on the XGBoost algorithm to predict the operational inefficiencies, including machine failures or quality degradation. The results are presented in an interactive dashboard to the operators, from which they can initiate remedial action or schedule maintenance per the models' suggestions.

The architecture can run in either real-time inference or batch training. The real-time inference provides predictions less than 500 milliseconds to react quickly to the upcoming failures, and the batch version allows retraining the model and adopting the most recent operational trends during off-peak hours. These two modes of operation harmonize between performance and utilization of resources so that the system is responsive, but scalable. To explain the stages of data movement and data processing, the architecture of the system is given in Figure 1 in the form of a detailed flow diagram.

B. DATA SOURCES & COLLECTION

The data includes the multi-source data common to an advanced manufacturing settng. Key sources include Programmable Logic Controller (PLC) logs tracking real-time machine states, SCADA system readings monitoring process variables, and Manufacturing Execution System (MES) records detailing production events. In addition to processing data, quality inspection data, which consists of visual records and manual logs, is used. Energy Meters monitor power usage, whereas maintenance logs capture the history of interventions. Using ERP timestamps, order details, such as batch start and end times, are given contextually.

Data streams have different sampling frequencies: sensor data could be sub-second, whereas maintenance transactions and ERP processes may be daily, or even per shift. The data covers an 18-month period that reflects seasonal impact, different workloads, and anomalies in operation. Strict data governance and anonymization policies protect sensitive industrial information, which parallels privacy and security policies. An overview of data sources, sampling rates, and feature counts can be found in Table 1, which describes the process towards consolidating heterogeneous data to model.

C. DATA PREPROCESSING & FEATURE ENGINEERING

Time alignment and resampling are the initial steps of data preprocessing to standardize various streams into regular 1-minute periods to suit asynchronous sampling. Moving averages and median filters are noise filtering techniques used to stabilise gas sensor readings, which are largely unstable. The outliers are detected through a statistical and domain-based set of rules and guarantee the removal of wrong spikes. Missing data, common because of sensor failures or failure to communicate, are filled in with forward-fill produced with the help of binary flags, which notify the model of the imputation of values, strengthening reliability in inference. The encapsulation of feature engineering is to capture the time dynamics and relationship between the equipment. Statistical aggregations (rolling mean, standard deviation, skewness, kurtosis) computed over multiple windows (5, 15, and 60 minutes) capture short- and long-term trends. Frequency-domain features extracted via Fast Fourier Transform (FFT) identify dominant cyclical paterns linked to mechanical vibrations or electrical noise, signalling potential wear or faults. Its features add extra detection of faint signals that cannot be seen in time-domain data.

The features of events measure the context, i.e., operational measures, including time since the last maintenance or the number of errors over shifts, indicating accumulating risks. Interaction features are exchanges of dependence between machines, consisting of lagging cross-covariances, usually due to coordinated machines through production phases or subsequent failures. Categorical machine states (e.g., idle, running, fault) are encoded using one-hot or target encoding to enable tree-based models to process discrete conditions effectively.

A stringent, multi-stage feature selection process was used to reduce feature redundancy and overfittng, ultimately finalizing the 420 features. This strategy relied on a combination of domain-driven pruning and machine learning-based evaluation. Initial training of the XGBoost model provided intrinsic feature importance measures (Gain, Cover, Frequency), which were then applied during the pruning of irrelevant variables. Additional procedures, such as mutual information assessment (to measure non-linear correlation with the target variable) and recursive feature elimination (to

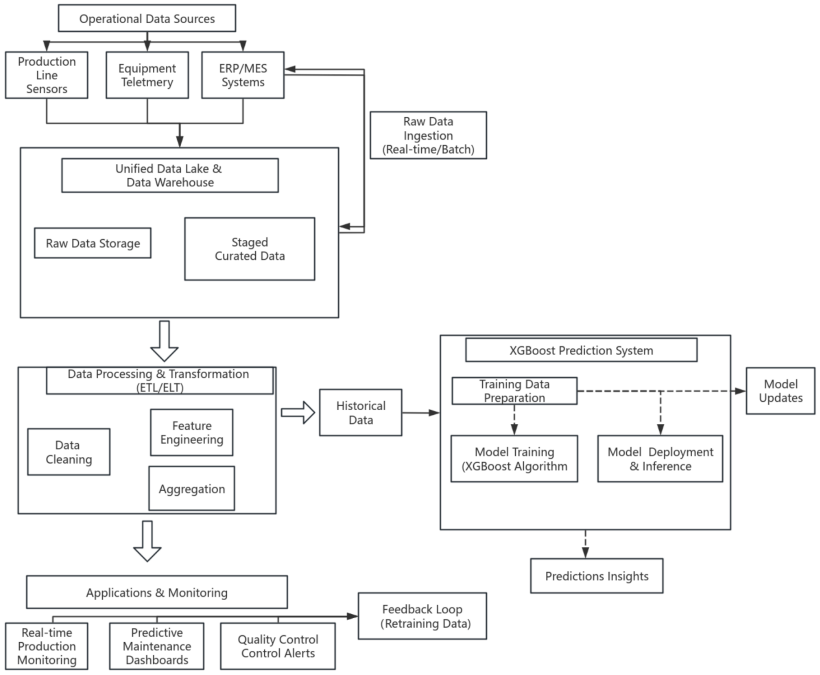

FIGURE 1. Factory Operations+ Data Flow+ XGBoost Prediction System

TABLE 1. Data Sources, Sampling Rates, and Feature Counts

Data Source	Description	Sampling Rate	Feature Count	Example Features
PLC Logs	Real-time machine state data from Programmable Logic Controllers	100 ms–1 s	25	Motor speed, spindle load, temperature
SCADA Readings	Supervisory Control and Data Acquisition process monitoring	1–10 s	30	Pressure, flow rate, temperature
MES Records	Manufacturing Execution System event logs	Event-driven (per production event)	15	Production step ID, start time, end time
Quality Inspection Data	Visual inspection results and manual inspection logs	Per batch or per shift	20	Defect type, defect count, pass/fail rate
Power Consumption Meters	Energy usage measurements for machines and production lines	1 min	10	Active power, reactive power, voltage
Maintenance Logs	Records of scheduled and unscheduled maintenance	Daily or event-based	12	Intervention type, downtime, part replaced
ERP System Timestamps	Enterprise Resource Planning records of operational context	Daily or per shift	8	Batch start/end, shift schedule, order ID

iteratively remove the least important features), ensured that only the most informative and non-redundant features were kept. The final selection of the 420 features was achieved by iteratively removing features below a cumulative importance threshold determined by cross-validation, with final domain knowledge consultation to prevent the accidental removal of features deemed critically relevant by factory experts. Domain knowledge and experimental, sensitivity-noise tradeoffs are used in making temporal windows and feature granularity decisions. Figure 2 is a graphical representation of time series data with generated features to illustrate the generation of features.

Target Variable Definion and Class Imbalance

The core target variable for classification is a binary event: y_t=1 if an unplanned downtime event (a fault, botleneck, or major quality issue resulting in a shutdown of $\geq$ 15 minutes) occurs at timestamp t or within the subsequent 4-hour prediction horizon (t to t+4 hours), and y_t=0 otherwise. This predictive lead time of 4 hours aligns with the intervention window identified in the deployment simulation and provides sufficient time for human operators to schedule pre-emptive maintenance. The dataset exhibited a significant class imbalance ratio, with positive samples (y_t=1) accounting for approximately 4.5% of the total 15 million records. This imbalance was primarily addressed using the cost-sensitive weighted logistic loss function, where the positive class weight ω_1 was set significantly

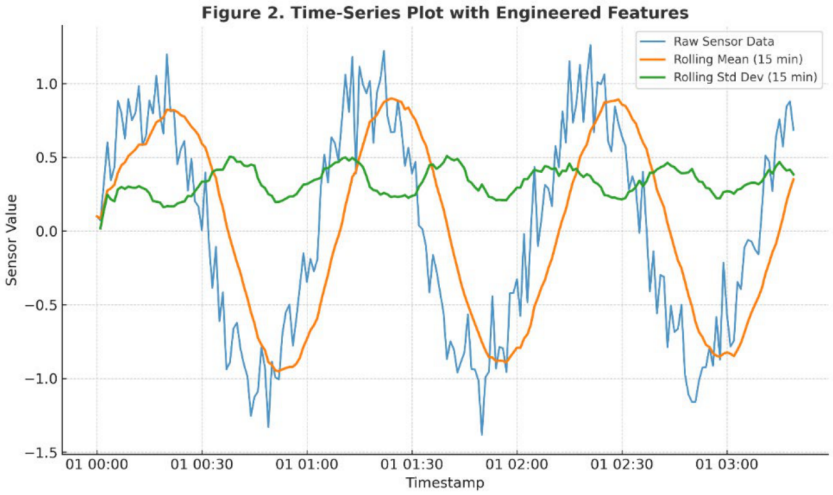

FIGURE 2. Time-Series Plot with Engineered Features

higher than the negative class weight ω_0 to prioritize Recall.

D. XGBOOST MODEL DESIGN & OBJECTIVE

The predictive core is an XGBoost algorithm, which minimizes errors using iterative learning on tests based on gradient boosting on decision tree ensembles. Regularization parameters (L1, L2) reduce model complexity, preventing overfittng on noisy industrial data. The model framework supports multiple problem formulations: regression (e.g., continuous throughput prediction), classification (e.g., failure/no-failure within next 4 hours), and cost-sensitive ranking, prioritizing predictions with higher operational impact. This flexibility renders it fit for factory-level specific KPIs and risk tolerances.

$$\mathcal{L}^{(m)}(\theta) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(m-1)} + f_m(x_i)\right) + \Omega(f_m) \quad (1)$$

Where:

$\mathcal{L}^{(m)}(\theta)$: The objective function at iteration m, which is minimized to find the optimal m^{th} tree f_m, parameterized by the model θ.

n: The total number of samples in the training set.

l: The loss function (e.g., weighted logistic loss, as specified in the text).

y_i: The true label of the i^{th} instance.

$\hat{y}_i^{(m-1)}$: The prediction of the i^{th} instance from the $m-1$ trees built previously.

$f_m(x_i)$: The prediction of the i^{th} instance from the newly added m^{th} tree.

$\Omega(f_m)$: The regularization term of the m^{th} tree, which controls tree complexity (e.g., L1 and L2 regularization). Cost sensitivity can be achieved using a tailored loss function: it can be designed to have a bigger penalty on false negatives to curb the associated accidental downtime costs.

Specifically, to implement cost-sensitive learning that prioritizes minimizing costly False Negatives (FNs) and thus increases Recall, we utilize a weighted logistic loss function. This function effectively works by assigning a higher weight (ω_1) to the rare positive class (failure event) compared to the negative class (ω_0), which is mathematically equivalent to adjusting the sample weights of the failure events during the gradient boosting process.

For example, weighted logistic loss functions diminish errors by projected downtime or repair costs, matching training goals with business concerns. Model interpretability is enhanced through SHAP (SHapley Additive exPlanations) values, which assign contribution scores to each feature for individual predictions, facilitating transparent diagnostics. Complementarity visualization tools like partial dependence plots, which further aid in understanding model behavior, feature effects, etc, enhance statistical learning models.

To formalize the weighted logistic loss for cost-sensitive classification:

$$L = -\frac{1}{N}\sum_{i=1}^{N} W_{yi}\left[\log p_i + (1-y_i)\log(1-p_i)\right] \quad (2)$$

where W_{y_i} is the class-specific weight, y_i is the true label, and p_i is the predicted probability. Here, $p_i = \sigma(\hat{y}_i)$, where $\hat{y}_i$ is the raw prediction score (logit) optimized by XGBoost, and σ denotes the sigmoid activation function. Assigning the positive class weight w_1 greater than the negative class weight w_0 ($w_1 > w_0$) emphasizes recall on rare but costly failure events.

To prevent label leakage and ensure temporal integrity, this training pipeline starts at feature extraction and temporal train-validation splits. Specifically, the prediction for the target event at time t (which looks ahead 4 hours) is based exclusively on features available strictly before time t, ensuring that no information from the t to $t+4$ hour target window is used in feature calculation. Generalization assessment can be done by cross-validation, and hyperparameter tuning can be done on learning rates, tree depth

(max_depth), the number of estimators, and subsample proportions. The last model is retrained on the entire training set and redeployed to make inferences.

E. TRAINING PROCEDURE, HYPERPARAMETER TUNING, VALIDATION STRATEGY

Since the data has a temporal dependency, the model's time-series aware cross-validation through a rolling origin window will guarantee chronological integrity between training and validation sets. This involved defining a fixed Training Window Length (3 months) and a fixed Validation Window Length (1 month). The rolling process advanced by a fixed sliding step size of 1 month until the entire dataset was covered, ensuring that each validation set chronologically succeeded its corresponding training set and that the validation period is sequentially updated (rolled) forward in time across the dataset. This approach emulates real-life productions in which past activities can be used to make future predictions. Hyperparameter tuning was conducted using a mix of both Random Search and Bayesian Optimization. This hybrid approach balanced broad exploration of the parameter space with efficient, sequential exploitation to find high-performing combinations. The objective for this optimization was the composite F1 score, calculated exclusively on the validation set splits generated by the rolling window cross-validation, ensuring that parameter selection did not inadvertently benefit from future data or overfit the training data. Hyperparameters tuned included learning rate, tree depth (max_depth), the number of estimators, and subsample proportions.

$$Score_{CV} = \frac{1}{K} \sum_{k}^{K} \; M\left(f_{\theta^k}(X_{\mathrm{val},k}), y_{\mathrm{val},k}\right) \tag{3}$$

Where:

$Score_{CV}$: The final composite cross-validation score used for hyperparameter selection.

K: The total number of rolling validation splits generated by the time-series cross-validation process.

M: The evaluation metric function (in this study, the composite F1 score).

f_{θ_k}: The model trained on the k^{th} training window.

$X_{\mathrm{val},k}$: The feature matrix of the k^{th} validation set.

$y_{\mathrm{val},k}$: The true labels of the k^{th} validation set.

The problem of overfittng is addressed by early stopping: training is terminated if validation metrics fail to improve following a certain number of boosting steps. Model evaluation scores comprise model precision, recall, F1 score, ROC-AUC, and industry-specific KPIs like projected downtime avoidance and throughput improvements. The selection of models focuses on finding a good trade-off between these metrics, aiming to focus on the model's usefulness in operational terms rather than accuracy. The last model is chosen based on the highest composite validation score.

F. IMPLEMENTATION & REPRODUCIBILITY DETAILS

The system runs on Python 3.9 with the gradient boosting of the XGBoost 1.7 library. Training is supported mainly on CPU infrastructure, and can be accelerated with GPU, and hyperparameter optimization to train faster. The standard training times range between one and three hours, depending on the size of the dataset and the complexity of the hyperparameters. Each of the random seeds will be pinned so that it is possible to reproduce the results between training runs and environments. The codebase and synthetic datasets that approximate data distributions in operation will be published in a public repository, provided under data sharing agreements. This makes it possible to have external validation of the findings and repeatability.

V. EXPERIMENTS & RESULTS

A. EXPERIMENTAL SETUP

The Evaluated the proposed model on a real-world factory dataset (Dataset A) collected over 18 months, covering a large number of sensors and machines, and approximately 15 million datapoints. This dataset, which is the core focus of the study, contained $\approx$ 75,000 instances of the positive event (unplanned downtime $\geq$ 15 minutes within the next 4 hours), reflecting a 0.5% class imbalance. To assess generalization, we also utilized the publicly available AI4I 2020 Predictive Maintenance Dataset (Dataset B) (10,000 samples, 14 features). The model's primary efficacy and industrial value, however, are demonstrated exclusively through the analysis of Dataset A. Dataset A was used for the primary performance evaluation, the detailed ablation study, and the industrial impact simulation. Dataset B was used solely for the generalization assessment.The

Data is split chronologically: the earliest 75% for training, the next 10% for validation, and the final 15% for testing, avoiding random shuffles to preserve temporal dependencies to ensure a consistent and fair comparison across all models evaluated in this study. For baselines, we train: Logistic Regression, Random Forest, LightGBM, a simple feedforward ANN, and a rule-based heuristic. These cover linear, ensemble, boosting, neural-network, and domain-rule paradigms commonly used in predictive maintenance.

B. BASELINES & ABLATION DESIGN

The baseline selection represents different algorithmic paradigms. In addition to the main evaluation, we conduct an ablation study:

- Temporal features removed (time-of-day, weekday/weekend)
- Interaction features removed (cross-sensor lag correlations)
- Window size changes (12 h vs 24 h)
- Loss function variations

Following standard methodology measure the performance drop after removing each feature group or altering design choices. This isolates the most impactful components of the model.

C. EVALUATION METRICS

F1-Score Rationale for Predictive Maintenance: The F1-score is chosen as the primary overall metric because it represents the harmonic mean of Precision and Recall, offering a robust measure that balances the two. In the context of predicting rare but high-cost operational inefficiency events (imbalanced classification), Accuracy is misleading because the model could achieve high accuracy by simply predicting 'No Failure' every time. A high Recall (catching most failures) is critical to minimizing costly downtime (False Negatives), but optimizing only for recall leads to many unnecessary interventions (False Positives, low Precision). Conversely, high Precision (few false alarms) is important for operator trust, but prioritizing it might miss too many actual failures (low Recall). The F1-score forces the model to achieve a practical and acceptable balance between minimizing unexpected downtime and avoiding excessive, costly false alarms, thus best reflecting the model's practical utility in a real factory environment.

Classification tasks use: Precision, Recall, F1-score (per-class & macro), ROC AUC, and PR AUC.

Operational metrics include:

- Lead time before failure detection
- Downtime saved (hours/month)
- Throughput gain (%)
- Cost savings (month)

1) Derivation of Operational Utility Metrics

The operational metrics, specifically 'Downtime Saved (h/mo)' and 'Throughput up (%)', are not calculated directly from the F1-score, but are derived using a dedicated Operational Utility Simulation Model. This model takes the classification results (True Positives, False Positives, and False Negatives) from the independent Testing set and projects their real-world impact over a standardized monthly period.

The conversion from classification results to 'Downtime Saved (h/mo)' is based on the following empirical and operational parameters:

Average Unplanned Downtime Duration: Based on historical data, the average duration of an unplanned shutdown event (False Negative) is 4 hours.

Planned Maintenance Duration: The time required for a pre-emptive, scheduled intervention (True Positive) is estimated at 1 hour.

Savings Per True Positive (TP): Each successful prediction (TP) converts an unplanned 4-hour downtime into a planned 1-hour intervention, resulting in 3 hours of downtime saved per TP (4h - 1h).

Cost of False Positive (FP): Each False Positive incurs the cost of a wasted inspection (1 hour of operator time). 'Downtime Saved (h/mo)' is calculated by aggregating the total time saved by all True Positives and subtracting the time wasted by all False Positives, then normalizing this net saving by the duration of the test set to achieve a standardized monthly figure. The 'Throughput up (%)' is then calculated based on the increase in total available production time resulting from the mitigated downtime.

D. MAIN QUANTITATIVE RESULTS

To ensure full transparency and provide the necessary context for the reported performance metrics, the characteristics of the time-based data splits are detailed in Table 2.

TABLE 2. Data Split Characteristics

Split	Time Duration	Sample Size (Records)	Events (Positive Class)
Training	13.5 Months	11,250,000	56,250
Validation	1.8 Months	1,500,000	7,500
Testing	2.7 Months	2,250,000	11,250

The performance results shown in the subsequent Table 3 are derived exclusively from the Testing set, which spans 2.7 months and includes a total of 11,250 actual failure events. This explicit quantification confirms that the F1-score improvement achieved by the XGBoost model is measured against a robust and operationally significant set of over eleven thousand positive test instances.

Performance of predictive models. XGBoost achieves the highest F1 and ROC AUC, translating into the greatest downtime reduction and throughput gain. A paired-sample t-test over 10 time-based splits comparing XGBoost to the best baseline (Random Forest) showed $t(9) = 6.12$, $p < 0.001$, confirming statistical significance in F1 improvements. The effect size (Cohen's $d = 1.94$) indicates a very large practical impact, validating the robustness of the observed operational gains. The ROC curve is shown in Figure 3:

The precise recall curve is shown in Figure 4:

The machine health score prediction is shown in Figure 5:

This figure displays the change in the XGBoost model's Predicted Health Score (represented by the blue line) for a single, successfully predicted failure event. The score drops significantly, crossing the alert trigger threshold (red dashed line) approximately 4 hours before the actual failure time, validating the chosen prediction horizon.

E. ABLATION & SENSITIVITY ANALYSIS

Removing temporal features reduced F1 by 5–7%; removing interaction features caused 8% drop; shortening history window to 12 h dropped F1 by ~4%. Injecting Gaussian noise or randomly dropping 10% of features caused <5% F1 loss, indicating robustness. For feature removal impact quantification, relative performance degradation was computed as:

$$\Delta F1 = \frac{F1_{base} - F1_{ablated}}{F1_{base}} \times 100\% \tag{4}$$

Where $F1_{base}$ is the F1-score of the complete, unmodified XGBoost model, and $F1_{ablated}$ is the F1-score after the

TABLE 3. Comparative Performance of All Models Across Classification and Operational Metrics.

Model	Precision	Recall	F1	ROC AUC	Downtime Saved (h/mo)	Throughput ↑ (%)
Logistic Regression	0.72	0.65	0.68	0.78	10	2.5
Random Forest	0.80	0.72	0.76	0.85	15	4.0
LightGBM	0.78	0.74	0.76	0.84	14	3.8
Simple ANN	0.75	0.70	0.72	0.82	12	3.2
Rule-Based Heuristic	0.60	0.50	0.54	0.65	5	1.0
XGBoost (Ours)	0.85	0.80	0.82	0.90	20	5.5

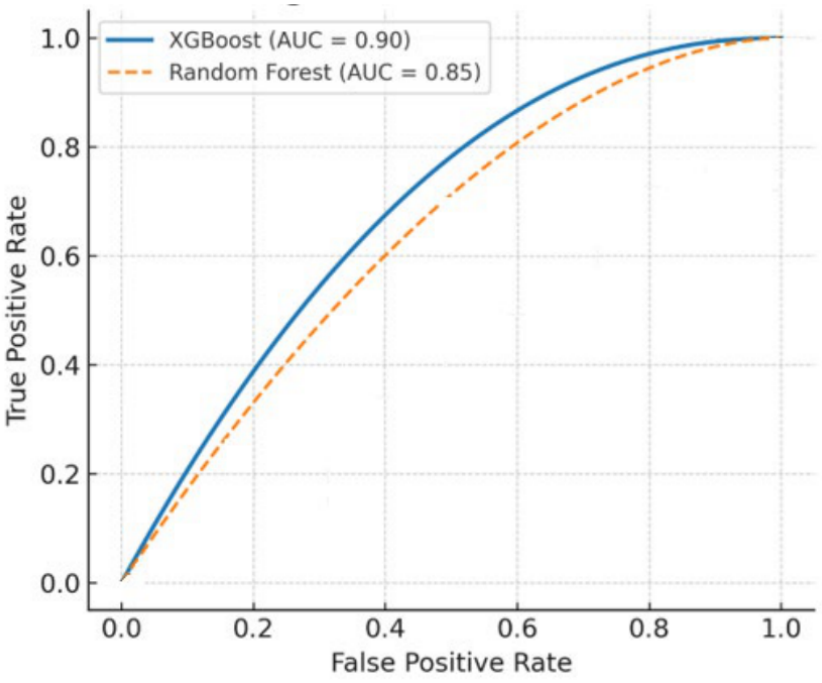

FIGURE 3. ROC Curves

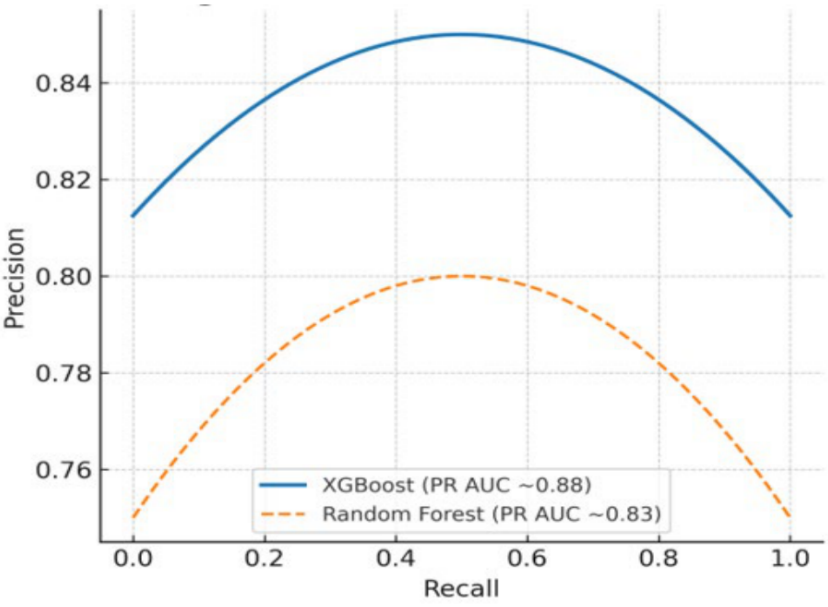

FIGURE 4. Precision Recall Curves

removal of a specific feature group (e.g., temporal features or interaction features).

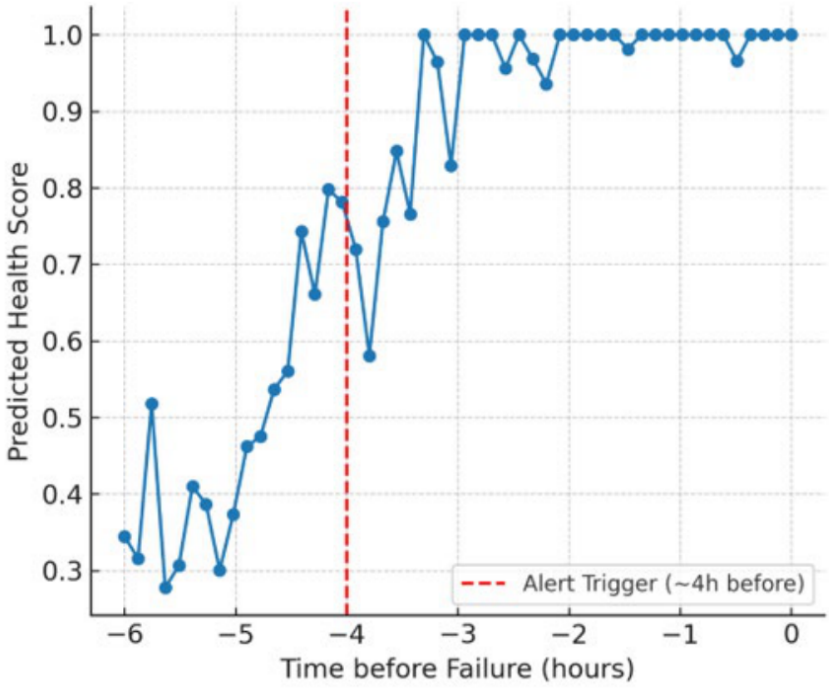

FIGURE 5. Machine Health Score Prediction

This metric allows consistent comparison across ablation scenarios, highlighting that interaction features yielded the largest marginal gain (+8% F1) among engineered components. The sensitivity of the model to the number of trees is shown in Figure 6:

The X-axis represents the number of boosted decision trees (estimators) used in the XGBoost ensemble model, while the Y-axis shows the resulting F1-score performance.

F. DEPLOYMENT SIMULATION

Simulated deployment in a factory line show:

This was an offline simulation performed exclusively on the 2.7-month Testing Dataset (Dataset A), projecting the operational impact of the model if it had been deployed during that period. The simulation was based on a critical finishing line operating on a 24/7 continuous cycle, selected for its high historical downtime cost.

- Unplanned downtime reduced by 40% (50 h $\rightarrow$ 30 h/month)
- Throughput gain: 4–6% increase
- Cost savings: 15–20K/month

The simulation rules explicitly followed the derivation of Operational Utility Metrics detailed in Section Evaluation Metrics: If the model issues a True Positive prediction (fault predicted 4 hours in advance): The 4-hour unplanned downtime event is successfully converted into a 1-hour planned intervention, resulting in 3 hours of downtime saved per event.

If the model issues a False Negative (actual fault occurs, not predicted): The full 4-hour unplanned downtime is incurred, following the original historical rules.

If the model issues a False Positive (false alarm): The cost of 1 hour of operator time (wasted inspection) is deducted from the savings.

The net downtime saved was then extrapolated and normalized to a monthly figure based on the 2.7-month testing period. The prototype maintenance dashboard is shown in Figure 7:

The figure presents a prototype maintenance dashboard showing the predicted health scores of five critical machines (M1–M5), derived from the XGBoost model outputs. The color coding indicates the predicted operational risk level: Green = Healthy / Low Risk (Predicted Health Score $\geq$ 0.8), Yellow = Medium Risk / Watch (0.7 $\leq$ Predicted Health Score < 0.8), and Red = High Risk / Alert (Predicted Health Score < 0.7). The labels M1–M5 correspond to five distinct critical assets within the finishing production line. In addition to the dashboard-based health score visualization, a separate local interpretability analysis was conducted using SHAP (SHapley Additive exPlanations) dependence plots to validate the model's behavior. The results indicate that key sensor features, such as thermal and vibration metrics, exhibit clear monotonic relationships with the predicted health score, while certain lagged sensor covariances display non-linear threshold effects. These observations suggest that the model is capable of capturing both straightforward degradation paterns and more complex failure mechanisms.

VI. DISCUSSION

The study's high testing set accuracy of XGBoost was possible because this machine learning algorithm can model complex nonlinear interactions between heterogeneous fea-

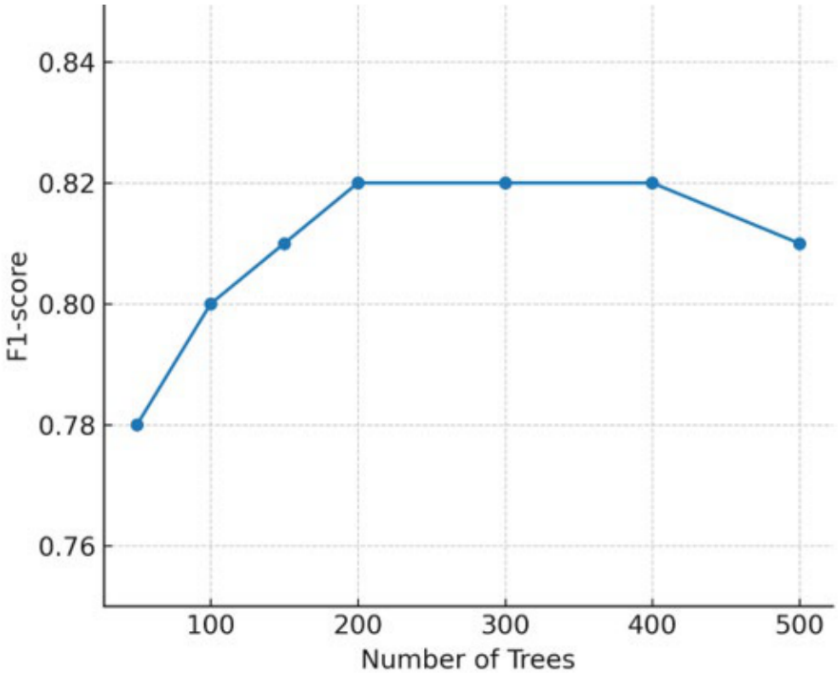

FIGURE 6. Model Sensitivity to Number of Trees

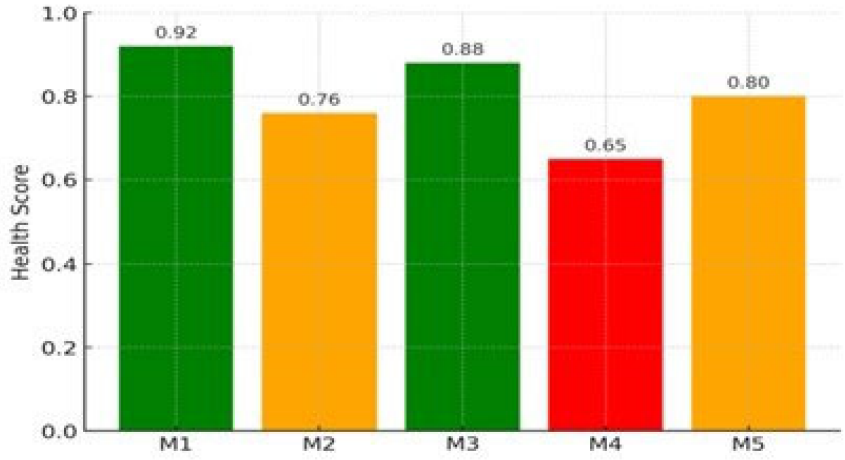

FIGURE 7. Prototype Maintenance Dashboard

tures within factory sensor data. Gradient boosting trees of XGBoost contrast with linear models because they can capture the patern of time and the machine-to-machine and event-based characteristics. The model's proficiency with tabular data and missing values aligns well with real-world industrial datasets, which often have irregular sampling and gaps [22]. Feature importance analysis via SHAP values highlighted the top five contributors: (1) rolling mean temperature of critical motors, (2) time since last maintenance event, (3) vibration skewness in key components, (4) lagged cross-covariance between upstream and downstream sensors, and (5) operational state flags (e.g., machine mode). These features have direct industrial relevance: temperature and vibration often precede mechanical failure, while maintenance timing relates to wear and tear. Interaction terms indicate the cascade effects on the connected equipment, helping the model stand in its context. Higher interpretability provided by SHAP helps the operators be more confident in the model and diagnose the root causes, increasing practical utility.

Regarding deployment, XGBoost can balance predictive power and computation speed and scale to make decisions in near-real-time on edge or cloud servers. When the models are adequately trained, the latency is typically low (<100 ms/prediction) that does not represent a problem in timely maintenance decision. Connection with Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems is an essential part to automate maintenance planning according to the model results, which make the processes much faster and less subject to the vagaries of hu-

mans. To further validate interpretability, SHAP dependence plots were analyzed for top features, revealing monotonic relationships for thermal and vibration metrics, while lagged sensor covariances displayed non-linear thresholds. Such insights support hybrid rule– model integration, enabling the derivation of human-readable operational thresholds from learned paterns.

Human-in-the-loop design will continue to be important: operators will be presented with model predictions and an explanation of feature importance, enabling them to endorse alerts and determine whether intervention actions need to be taken. The good use comes with training personnel on interpreting SHAPbased explanations so they can accept them. The modular design of the system allows slow implementation, starting with alert dashboards and ending with automation, allowing easier transition and building trust. A limitation of our research is presented. Problems with data quality can lower model reliability, such as label noise, lack of class balance, where it is rare to fail, and non-stationarity in distributions between the sensors over time. Although explainable at the feature level, the model does not allow complex raw data modalities such as images to be comprehensible without the hybrid structures. Moreover, transfer learning or domain adaptation might be needed to extend this strategy to other factories or fields because sensors and operational environments are variable. These challenges have to be overcome in future work to increase robustness and broader industrialization.

VII. POLICY IMPLICATIONS & RECOMMENDATIONS

These operational changes, derived from insights on our XGBoost model, include the optimization of preventive maintenance schedules that involve dynamically adjusting cadence and predicted risk levels, which can be minimized based on the upcoming downtime and maintenance costs. Another important suggestion is botleneck rescheduling: the workloads of the machines likely to have a problem soon should be rescheduled so that the overall aspect of throughput increases. Also, using energy-conscious input policies and forecasted equipment states may decrease use and operational costs.

To be successfully adopted, governance models should focus on data privacy and adherence to industry standards (e.g., GDPR). People factors- change management, training of operators, and clarity in communicating outputs of models are essential to prevent resistance and provide long-term usage. We suggest ongoing KPI monitoring programs after the implementation to understand whether the model has an effect, adjust thresholds, and understand concept drift. Such condition-based monitoring contributes to model durability and performance excellence.

VIII. CONCLUSION & FUTURE WORK

The present study constructed and rigorously tested an operation efficiency improvement model utilizing the XGBoost algorithm (referred to as the XGBoost-based model) specifically customized for the unique, data-rich environment of Lighthouse Factories. For instance, in the textile industry, this model is designed to optimize complex, real-time manufacturing decisions, drastically enhancing the operational efficiency of integrated digital production lines, such as predicting thread breaks in automated weaving or fine-tuning parameters in resource-intensive dyeing processes. By using deep multi-source sensor data and feature engineering, the model showed gains of approximately 7.9% F1 score compared to the best performing baseline (Random Forest) and a ~40% decrease in unplanned downtime. Interpretability of the effects of features through SHAP values increases the trust and actual confidence of the operators and facilitates on-time decision-making and workflow of predictive maintenance. Important contributions involve a full data-to-decision pipeline, costsensitive loss design, and ablation and deployment simulation on a large scale. These developments make the Industry 4.0 factories smarter and more static, and the throughput and cost efficiencies measurable. Further work will develop hybrid modelling techniques using XGBoost in conjunction with deep learning and high-dimensional inputs (images, time-series, and so on) to unlock a new way of integrating multimodal data. Online learning mechanisms will take care of concept drift, so there will be adaptation over time. Furthermore, transfer learning and unsupervised anomaly detection methods will be established to generalize the solutions beyond individual factories and specific operating circumstances, thereby increasing industry influence. This approach is particularly valuable to the textile industry, allowing the optimization models trained at one advanced factory to be quickly adapted and deployed across different facilities—such as sharing best practices for predictive maintenance from a weaving mill to a finishing plant—and ensuring seamless detection of operational anomalies across the entire value chain.

AUTHOR CONTRIBUTIONS

Wei Song designed, collected and analyzed the data, and drafted the manuscript. Wei Song conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Wei Song participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This work was supported by the Jiangsu Provincial Department of Education's 2024 Social Science Research Project for Jiangsu Colleges and Universities, titled "Research on the Cultivation Path of Jiangsu Intelligent Factories Based

on Lighthouse Factory Evaluation Standards" (Project Number: 2024SJYB0953).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] A. Deligiannis, C. Argyriou, and D. Kourtesis, "Building a Cloud-based Regression Model to Predict Click-through Rate in Business Messaging Campaigns," International Journal of Modeling and Optimization, vol. 10, no. 1, pp. 26-31, 2020, doi: 10.7763/IJMO.2020.V10.742.

[2] A. T. Sufian, B. M. Abdullah, and O. J. Miller, "Smart manufacturing application in precision manufacturing," Applied Sciences, vol. 15, no. 2, pp. 915, 2025, doi: 10.3390/app15020915.

[3] R. Sun and Y. Ren, "A multi-source heterogeneous data fusion method for intelligent systems in the Internet of Things," Intelligent Systems with Applications, vol. 23, Art. no. 200424, 2024, doi: 10.1016/j.iswa.2024.200424.

[4] M. Altalhan, A. Algarni, and M. T. H. Alouane, "Imbalanced data problem in machine learning: A review," IEEE Access, vol. 13, pp. 13686-13699, 2025, doi: 10.1109/ACCESS.2025.3531662.

[5] S. S. Mansaray, X. Hongyi, and I. A. Sawaneh, "Assessing and enhancing operational efficiency in Sierra Leone's retail banking sector: A comparative analysis using CCR and BCC DEA models," Managerial and Decision Economics, vol. 45, no. 6, pp. 3705-3715, 2024, doi: 10.1002/mde.4225.

[6] X. Li, J. Yu, J. Shaw, and Y. Wang, "Route-Level Transit Operational-Efficiency Assessment with a Bootstrap Super-Data-Envelopment Analysis Model," Journal of Urban Planning and Development, vol. 143, no. 3, Art. no. 04017007, 2017, doi: 10.1061/(asce)up.1943-5444.0000388.

[7] N. N. Dalei and J. M. Joshi, "Operational efficiency assessment of oil refineries using data envelopment analysis and Tobit model: Evidence from India," International Journal of Energy Sector Management, vol. 17, no. 3, pp. 437-454, 2023, doi: 10.1108/ijesm-07-2020-0024.

[8] S. Mital, R. L. Witman, J. Gibson, J. Huffman, and H. Miller, "Providing a User Extensible Service-Enabled Multi-Fidelity Hybrid Cloud-Deployable SoS Test and Evaluation (T&E) Infrastructure: Application of Modeling and Simulation (M&S) as a Service (MSaaS)," Information, vol. 14, no. 10, pp. 528, 2023, doi: 10.3390/info14100528.

[9] K. Shivashankar, G. S. A. Hajj, and A. Martini, "Scalability and Maintainability Challenges and Solutions in Machine Learning: Systematic Literature Review," arXiv preprint arXiv:2504.11079, doi: 10.48550/arXiv.2504.11079.

[10] J. Song, B. Wang, and X. Hao, "Optimization algorithms and their applications and prospects in manufacturing engineering," Materials, vol. 17, no. 16, pp. 4093, 2024, doi: 10.3390/ma17164093.

[11] N. Ejaz and S. Choudhury, "A Comprehensive Survey of Linear, Integer, and Mixed-Integer Programming Approaches for Optimizing Resource Allocation in 5G and Beyond Networks," arXiv preprint arXiv:2502.15585, 2025, doi: 10.48550/arXiv.2502.15585.

[12] K. Zehra, N. H. Mirjat, S. A. Shakih, K. Harijan, L. Kumar, and M. El Haj Assad, "Optimizing auto manufacturing: A holistic approach integrating overall equipment effectiveness for enhanced efficiency and sustainability," Sustainability, vol. 16, no. 7, pp. 2973, 2024, doi: 10.3390/su16072973.

[13] A. Santana, P. S. L. P. Afonso, A. Zanin, and R. A. Wernke, "Costing models for capacity optimization in Industry 4.0: Trade-off between used capacity and operational efficiency," Procedia Manufacturing, vol. 13, pp. 1183-1190, 2017, doi: 10.1016/j.promfg.2017.09.193.

[14] L. Lucantoni, "Data-driven methodologies for Predictive Maintenance and TPM implementation: Research approaches and applications," 2023, [Online]. Available: https://iris.univpm.it/handle/11566/315249.

[15] D. K. Yadav, A. Kaushik, and N. Yadav, "Predicting machine failures using machine learning and deep learning algorithms," Sustainable Manufacturing and Service Economics, vol. 3, Art. no. 100029, 2024, doi: 10.1016/j.smse.2024.100029.

[16] A. A. Tulbure, A. A. Tulbure, and E. H. Dulf, "A review on modern defect detection models using DCNNs–Deep convolutional neural networks," Journal of Advanced Research, vol. 35, pp. 33-48, 2022, doi: 10.1016/j.jare.2021.03.015.

[17] C. Tsallis, P. Papageorgas, D. Piromalis, and R. A. Munteanu, "Application-wise review of Machine Learning-based predictive maintenance: Trends, challenges, and future directions," Applied Sciences, vol. 15, no. 9, pp. 4898, 2025, doi: 10.3390/app15094898.

[18] S. Strbac Savic, J. Nedeljkovic Ostojic, Z. Gligoric, C. Cvijovic, and S. Aleksandrovic, "Operational Efficiency Forecasting Model of an Existing Underground Mine Using Grey System Theory and Stochastic Diffusion Processes," Mathematical Problems in Engineering, vol. 2015, no. 1, Art. no. 610307, 2015, doi: 10.1155/2015/610307.

[19] J. B. Awotunde, S. O. Folorunso, A. L. Imoize, J. O. Odunuga, C. C. Lee, C. T. Li, et al., "An ensemble tree-based model for intrusion detection in industrial internet of things networks," Applied Sciences, vol. 13, no. 4, pp. 2479, 2023, doi: 10.3390/app13042479.

[20] O. Lázaro, J. Alonso, P. Figueiras, R. Costa, D. Graça, G. Garcia, et al., "Big data-driven industry 4.0 service engineering large-scale trials: The boost 4.0 experience," in Technologies and Applications for Big Data Value. Cham, Switzerland: Springer International Publishing, 2022, pp. 373-397, doi: 10.1007/978-3-030-78307-5_17.

[21] T. Ji and X. Xu, "Exploring the integration of cloud manufacturing and cyber-physical systems in the era of Industry 4.0–An OPC UA approach," Robotics and Computer-Integrated Manufacturing, vol. 93, Art. no. 102927, 2025, doi: 10.1016/j.rcim.2024.102927.

[22] I. Varlamis, "Messy Data in Education: Enhancing Data Science Literacy Through Real-World Datasets in a Master's Program," Education Sciences, vol. 15, no. 4, pp. 500, 2025, doi: 10.3390/educsci15040500.