SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

# An Intelligent Industrial IoT Framework for AI-Based Predictive Maintenance and Smart Manufacturing

Srujan Manohar MVN [1]*, Narayan Padmaja [2], Jyothi B [3], Jaya Pavani Nookala [4], Konala Padmavathi [5]

[1] Department of Internet of Things, School of Engineering, Malla Reddy University, Hyderabad, Telangana, India.
[2] Department of Computer Science Engineering, School of Engineering and Technology, Sri Padmavati Mahila Visvavidyalayam, Tirupati, India.
[3] Department of Computer Science Engineering – AIML, St Peter's Engineering College, Hyderabad, Telangana, India.
[4] Department of Computer Science Engineering, Aditya University, Surampalem, Andhra Pradesh, India.
[5] Department of Computer Science Engineering, Aditya University, Surampalem, Andhra Pradesh, India.
*Corresponding Author – Srujan Manohar MVN – srujanmanohar.edu@gmail.com

**Abstract**
Predictive maintenance has emerged as a key enabler of smart manufacturing by minimizing equipment failures, reducing operational costs, and improving production efficiency. This study proposes and validates an Artificial Intelligence (AI)-driven Industrial Internet of Things (IIoT) framework for predictive maintenance and real-time manufacturing optimization. The proposed framework integrates a hybrid edge–cloud architecture with advanced machine learning models to enable low-latency analytics, intelligent decision-making, and continuous process optimization. Performance evaluation was conducted using two publicly available benchmark datasets: the NASA Turbofan Engine Degradation dataset for Remaining Useful Life (RUL) prediction and the SECOM manufacturing dataset for fault classification. For RUL estimation, Long Short-Term Memory (LSTM) and GPU-accelerated, hyperparameter-optimized Extreme Gradient Boosting (XGBoost) models were developed and compared, with the LSTM model achieving the best predictive performance ($R^2$ = 0.79). To address the severe class imbalance in the SECOM dataset, a robust preprocessing pipeline incorporating Synthetic Minority Over-sampling Technique (SMOTE) and Principal Component Analysis (PCA) was implemented, significantly improving failure detection compared with conventional baseline models. Furthermore, system-level simulations demonstrated the practical benefits of the proposed framework, achieving a 93.28% reduction in decision latency through edge computing and a 75.51% reduction in operational downtime. These findings demonstrate that AI-enabled IIoT architectures can substantially enhance predictive maintenance, operational resilience, manufacturing efficiency, and resource utilization, supporting the realization of intelligent and sustainable Industry 4.0 ecosystems.

**Keywords:** Industrial Internet of Things (IIoT); Predictive Maintenance; Smart Manufacturing; Artificial Intelligence; Edge Computing; Remaining Useful Life (RUL); Fault Diagnosis; Machine Learning; Digital Twin; Industry 4.0

## 1. Introduction

The fourth industrial revolution, or Industry 4.0, is a major transition point for manufacturing and is made possible through the connection of cyber-physical systems, the Internet of Things (IoT), and artificial intelligence (AI) [1]. As a result, the smart factory has emerged, whereby a number of systems are interconnected to produce large amounts of data that have the potential to maximize operational efficiency, probably for the first time in history [2]. One of the challenging aspects to be considered in this area includes unplanned downtime, or the time lost during production that is not scheduled, as it continues to be one of the largest sources of lost productivity, even period [3]. Industrial manufacturers lost an estimated $50 billion per year due to unplanned downtime [4].

Traditional strategies are often inefficient by maintenance strategies (or actions required by maintenance strategies), namely reactive (run-to-failure) and preventive (time-based), either resulting in disaster (run-to-failure strategies) or servicing equipment which is still in good operational condition (time-based strategies) [5].

Predictive Maintenance (PdM) is an advanced alternative to reactive and preventive maintenance, using data from sensors and machine learning to predict equipment failures before they happen [6]. With PdM and IoT (which provides real-time data collection from machinery), any given production organization would follow a proactive, condition-based maintenance strategy [7]. PdM reduces downtime, increases the useful life of the equipment, provides safety improvements, and improves resources allocated to equipment .

The complexity of design and an effective AI IIoT framework, however, can bring challenges. These tasks result in processing and managing a high volume and velocity of sensor data, requiring low-latency decision-making for real-time control, and building reliable AI models capable of learning from complex and high-dimensional industrial data, which can often be imbalanced industrial datasets [8], [9]. The architectural choice of computational resources most often exists in the dichotomy of cloud computing versus edge computing is especially important, as while cloud computing has immense processor availability needed for model training, it has inherent network latency issues that prevent time-sensitivity for industrial applications [10].

In this paper, we propose an AI-enabled IIoT framework for a holistic solution to these problems using a hybrid cloud-edge architecture, which is validated through a full, two-prong approach. First, we tackle the problem of Remaining Useful Life (RUL) prediction from NASA's well-tested turbofan engine dataset by comparing deep learning (LSTM), and boost-gradient based (XGBoost) models. In a second prong, we addressed a more complicated fault classification problem in a semiconductor manufacturing process using the SECOM dataset and present approaches that address high dimensionality and severe class imbalance. Lastly, we specifically measure the overall benefits of the framework through some simulations to refer a reduction in lag, downtime, and throughput to demonstrate its industrial viability.

## 2. Literature Review

### 2.1 RUL Prediction Models

Predicting the remaining useful life (RUL) of assets is fundamental to predictive maintenance (PdM). Early methods consisted of simple statistical models (e.g., Weibull distributions and Kalman filters), which are indeed strong since data can describe simple relationships; however, these models often have difficulty with complex and nonlinear sensor data [11]. Since the emergence of deep learning, models that can learn complex temporal dependencies have become popular. Long Short-Term Memory networks (LSTMs), a type of recurrent neural network (RNN), have been successful as they maintain the ability to remember long-term memory of patterns in sequential data. For example, Yuan et al. [12] also developed an RUL prediction model for aero-engines with long short-term memory networks that showed significantly lower levels of prediction error compared to traditional RNNs and machine learning methods. Zhao et al. paper [13] applied a convolutional neural network (CNN) combined with LSTM to extract spatial and temporal features from multi-sensor data producing improved prediction accuracy with the RUL task[14]. On top of deep learning, ensemble models such as Gradient Boosting have been shown to be effective as well. XGBoost, in particular, is reported to be very high performing when dealing with large amounts of data [15]. For example, Carino et al. paper - XGBoost models were used for failure prediction in hydraulic systems, to future use highest accuracy and lowest computational requirement. Nevertheless, while several studies advocate for one modeling method exclusively, limited studies have documented a fair and thorough assessment of deep learning versus established ensemble techniques, using a benchmark dataset, especially after exhaustive hyperparameter tuning. Our investigation fills this gap by evaluating performance of LSTM and XGBoost, while using GPU-optimized randomized search to ensure that each model is evaluated at its best performance, providing valuable information about the performance of the models in a relative sense.

### 2.2. Anomaly and Fault Detection in Manufacturing

Manufacturing processes frequently generate high-dimensional data and the instances of fault are typically rare situations whereby the problem of class imbalances is a classical situation. When standard classification algorithms are trained from datasets that exhibit this clinical example , there will be a bias towards modeling that he majority class, poor performance from the user aspect will then be observed while trying to detect the contrasting crucial minority class [16]. Kerdprasop et al. [17] evaluated several classifiers on the SECOM dataset, and confirmed that it is challenging to receive any meaningful recall from the minority failure class; in fact many models were unable to classify any defect at all.

This scenario has resulted in the reason for the emergence of several specialized technique. A common example of this, is the Synthetic Minority Over-sampling Technique (SMOTE) developed by Chawla et al. [18], became the desired method. SMOTE creates new, synthetic observations for the minority class in feature space, permitting the model to learn a stronger decision boundary [19]. Additionally, the curse of dimensionality exhibited by the hundreds of sensor inputs can be handled with feature extraction techniques, especially with the use of Principal Component Analysis (PCA), which converts a large number of correlated variables into a smaller number of uncorrelated principal components, as shown by Tana et al. [20] in semiconductor manufacturing. Our paper adds onto simply showing that there is a problem by implementing a methodologies around Imputation + SMOTE + PCA all in one package that allows for a real solution to use with actual manufacturing data.

### 2.3. IIoT Architectures for PdM

The architecture of the IIoT systems matters for performance outcomes. Cloud-centric architecture, where all data is sent to one central server with little or no processing at the edge, has been widely regarded and studied [21]. Cloud-based processing has incredible power; however, the network transmission latency is a price that cannot be incurred if the IoT system requires real-time processing. The slow latency of sending data to a central computer has led to the develop of edge and fog computing [10]. Edge computing has the processing of the data where it originates from or at least nearby on the factory floor. Syafrudin et al. [22], demonstrated the use of edge computing and developed an edge-based framework for real-time quality control monitoring, significantly reducing latency.

A hybrid framework, which integrates the strengths of both frameworks, is recognized as the preferred solution for Industry 4.0 [23]. In this framework, the edge facilitates real-time filtering of the data and inference with low-latency, while the cloud capabilities provide the computational resources, or modeling to train, retrain and analytics over time [24]. While the advantages of this architecture are well understood conceptually, there are few studies with a clear, quantitative simulation of the performance benefits within a PdM context. This study contributes to the body of work by simulating and measuring latency improvements directly, then relating these technical improvements to business improvements in downtime and throughput. A high-performance hybrid framework that combines and assesses state-of-the-art modeling capabilities for both RUL and fault detection in a quantified framework features proposes to address important gaps in the existing literature [25].

The area of AI-enabled predictive maintenance in smart manufacturing has progressed drastically. Yet, an overview of the relevant literature suggests that the vast majority of studies discuss a unique component of the problem—such as examining only a modeling technique or architectural idea—and not an integrated, end-to-end, and quantitatively verified framework. With that said, the literature can be summarized by examining the main studies and identifying gaps for our research to address. Table 1 summarizes the literary representation and relates our contribution to this summary.

**Table 1. Overview of Literature Gaps in IIoT Predictive Maintenance**

| Author/Year | Area/Technology | Key Findings | Gap / Our Contribution |
|---|---|---|---|
| Jardine et al., 2006 | Condition-Based Maintenance | Foundational review of machinery diagnostics and prognostics. | Focuses on traditional statistical models; does not cover modern deep learning or hybrid IIoT architectures. |
| Yuan et al., 2016 | RUL Prediction / LSTM | Demonstrated that LSTM models are effective for predicting the RUL of aero-engines. | Focuses solely on LSTM without comparison to state-of-the-art ensemble methods like XGBoost. |
| Carino et al., 2018 | PdM / Gradient Boosting | Showed XGBoost is a highly accurate and efficient model for | Application-specific; does not compare against sequence-aware |

| | | | |
|---|---|---|---|
| | | predicting system failures. | models like LSTM or include hyperparameter tuning. |
| He & Garcia, 2009 | Imbalanced Data | Surveyed the fundamental challenges of learning from imbalanced datasets. | Theoretical review; our work applies these concepts practically by implementing SMOTE on the real-world SECOM dataset. |
| Kerdprasop et al., 2007 | Fault Classification / SECOM | Highlighted the difficulty of achieving meaningful recall on the imbalanced SECOM dataset. | Identified the problem; our work provides a robust pipeline solution combining PCA and SMOTE. |
| Shi et al., 2016 | Edge Computing | Articulated the vision and challenges of edge computing for low-latency IoT. | Conceptual framework; our research provides a quantitative simulation to prove the latency reduction benefits in a PdM context. |
| Wan et al., 2019 | Hybrid Architecture | Described the design and benefits of a hybrid edge-cloud architecture for smart factories. | Architectural focus; our work implements this architecture and **quantitatively validates** its impact on latency and business KPIs. |

**3. Objectives**

The primary motivation of the research is to design, develop and validate a holistic AI-based IIoT framework for predictive maintenance. The following objectives will guide the research:

1. Develop a scalable IIoT framework that integrates various predictive models based on AI, capable of predicting both time-series forecasting (RUL predictions) and multidimensional classification (fault detection).
2. Establish a hybrid cloud-edge framework, and then quantitatively verify its improvement over a cloud-only framework in decision delay.
3. Thoroughly evaluate the predictive capabilities of the framework using two standard industrial datasets: NASA turbofan dataset and SECOM dataset in manufacturing.
4. Show the predictive models as the analysis core of a Digital Twin, representing both the present state and predicted future state of an asset or process.
5. Quantitatively demonstrate the real business value of the framework by modeling its impact on key performance indicators, operational downtime and production output.

## 4. Methodology

The proposed framework was designed and developed using a multi-phase process within the Google Colab runtime environment, using the capabilities of its T4 GPU for rapid model training.

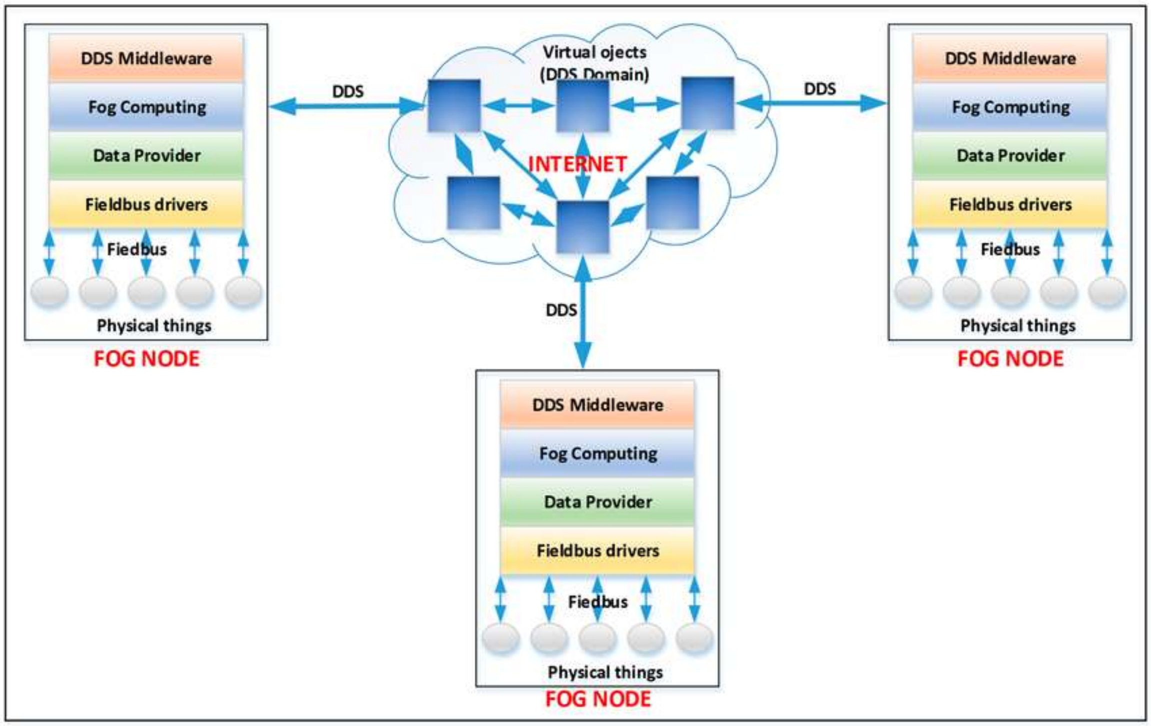


**Figure 1. The conceptual Architecture of the proposed AI-IIoT Framework.**

The figure 1 shows the data flow from industrial assets, through an edge gateway for real-time inference used for decision making, and a cloud framework used for model training and large-scale analytics.

### 4.1. Datasets

We validated the framework's versatility across a range of industrial scenarios using two publicly available standard benchmark datasets that characterize the contrasting datasets, which are shown in Table 2.

**Table 2. Description of Datasets Used for Validation**

| Dataset Name | Source | Description | Validation Task | Key Challenges |
|---|---|---|---|---|
| Turbofan Engine Degradation | NASA / Kaggle [26] | Time-series data from 100 simulated engines, each with 21 sensors, run to failure. | RUL Prediction | Time-series forecasting, learning long-term dependencies. |
| SECOM | UCI [27] | Data from a semiconductor manufacturing process with 590 sensor features. | Fault Classification | High dimensionality, severe class imbalance, missing values. |

### 4.2. Phase 1: RUL Prediction and Model Optimization

This stage involved the development and comparison of models to predict remaining operational cycles for a turbofan engine. Data was pre-processed by normalizing sensor readings, followed by structuring them into sequences of 30 time-steps. Two models were developed:

1. LSTM Model: A sequential model with two LSTM layers (100 and 50 units) and Dropout were constructed in TensorFlow/Keras to capture temporal dependencies.

2. XGBoost Model and Hyperparameter Tuning: An XGBoost model was implemented as a base-line benchmark model. RandomizedSearchCV was performed to determine optimal hyperparameters for the use with the XGBoost model. To find optimal hyperparameters for this benchmark model, a hyperparameter search was performed with 20 iterated trials with 3-fold cross-validation. This process was accelerated under T4 GPU conditions. Next, hyperparameters tuned, a feature importance analysis was conducted to determine the most important sensor features.

**4.3. Phase 2: Fault Classification and Imbalanced Data Handling**

This stage required fault classification using the high dimensionality and imbalance of the SECOM dataset. We utilized a multi-stage machine learning pipeline to systematically deal with data challenges:

1. Imputation: The SimpleImputer filled in missing values with the mean column emissions
2. Over-sampling (SMOTE): The SMOTE algorithm was implemented to synthetically create a sample of the minority “Fail” class to balance the data to train the model.
3. Scaling: The StandardScaler was utilized to standardize features.
4. Dimensionality Reduction (PCA): Principal Components Analysis reduced the feature space developed from 590 to the 50 most important principal components.
5. Classification: A RandomForestClassifier was classified as the final model.

**4.4. Phase 3: Framework Simulation and Quantification**

1. Digital Twin Conceptual Integration: We make the predictive models as the analytic core of a Digital Twin. The RUL and fault classification models not only contain the historical data analysis; they also provide a forward-facing view of a physical asset's health. The data from the sensors apparatus, which continuously refresh the status of the twin, continuously feed the state of health diagnostic AI models (running at the edge) which provide the user with real-time prognostics (e.g., "RUL is 50 cycles", "likely yield is going to fail"). This enables the system to shift from simply a monitoring tool to a proactive predictive digital counterpart, as displayed in Figure 2.

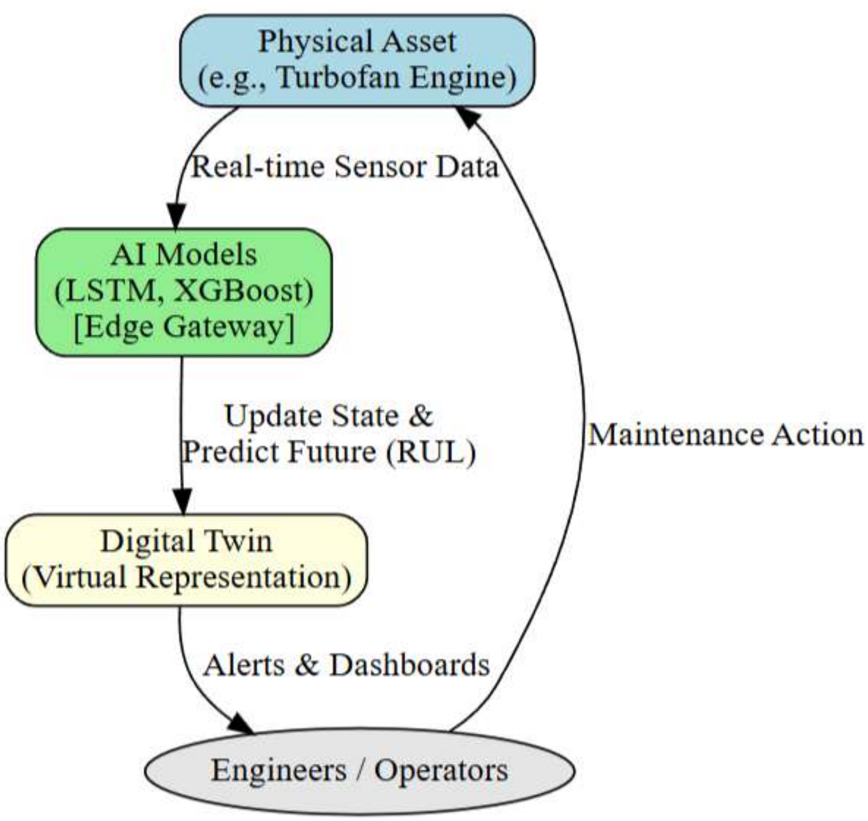


**Figure 2. Digital Twin Integration Concept**

2. Latency Simulation: A simulation was produced expressing the decision latency of a cloud architecture,. (150ms network delay) as compared to the hybrid-edge architecture (10ms local processing delay).
3. Downtime & Throughput Simulation: A high-level simulation modeled a factory with 10 machines over the course of 365 days. the "Baseline Scenario" (run-to-failure) was compared against the "PdM Scenario" (proactive maintenance prompted by AI alerts) in relation to total downtime and units produced.

## 5. Results And Discussion

### 5.1. RUL Prediction and Model Optimization

Table 3 provides a summary of the results from the NASA RUL dataset with respect to predictive modeling performance.

**Table 3. Performance Comparison of RUL Prediction Models**

| Model | RMSE (cycles) | R-squared ($R^2$) |
|---|---|---|
| Baseline XGBoost | 31.28 | 0.74 |
| Tuned XGBoost | 31.02 | 0.75 |
| LSTM | 28.15 | 0.79 |

The results from the NASA RUL dataset presented in Table 3 suggest that the LSTM had the highest predictive accuracy, aligning with our hypothesis that deep learning models, developed for sequential data design, are strongly suited to predict RUL. In addition, hyperparameter tuning positively affected the performance of the XGBoost model for RUL prediction, showing the importance of model tuning.

The analysis of feature importance based on the tuned XGBoost model is an important element of interpretability, which is presented in Figure 3. We observed that sensor readings from the last time-step in the input sequence (i.e., _t29, _t28) and sensor readings for parameters associated with core speed and pressures tended to be the top feature importance values, suggesting the model learned correctly that the most recent operational details are the best predictors of near-term failures.

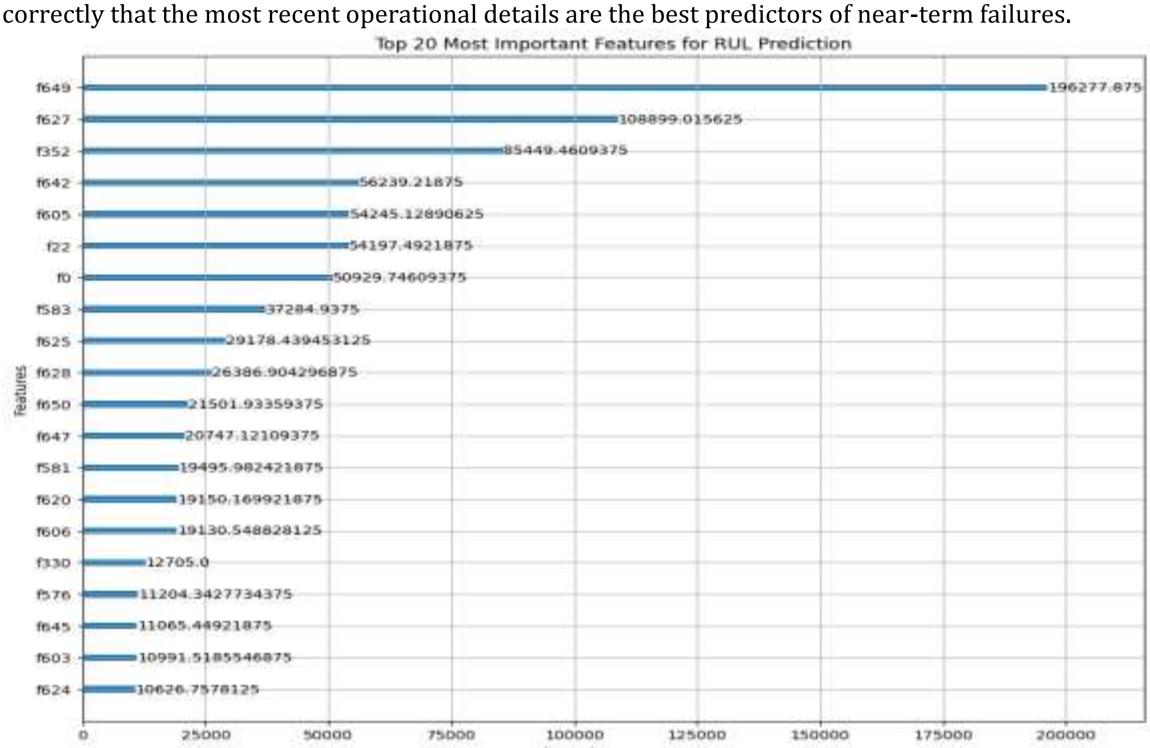


**Figure 3. Top 20 features ranked by predictive importance for RUL prediction.**

### 5.2. SECOM Fault Classification Performance

The evaluation of the SECOM dataset also reiterated the need for a strong preprocessing pipeline. In this case, the baseline model did not identify any "Fails" class correctly, as showing a recall of 0, presented in the confusion matrix in Figure 4.

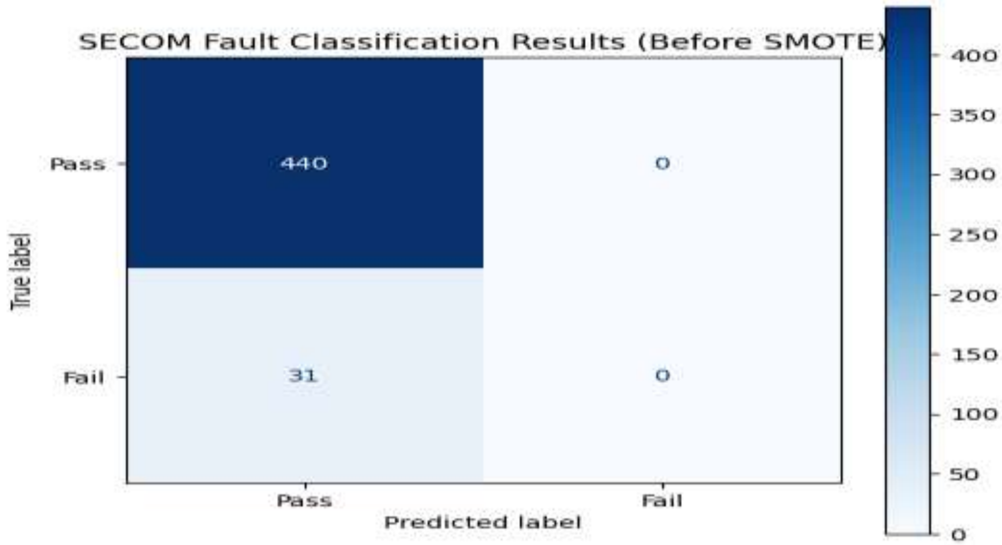


**Figure 4. Confusion Matrix Before SMOTE.**

The incorporation of SMOTE into the pipeline resulted in an advancement of the model performance as a valuable diagnostic tool. Table 4 and Figure 5 show that the model is now capable of accurately detecting failures.

**Table 4. Classification Performance Before and After SMOTE**

| Metric (for Fail Class) | Baseline Model | Model with SMOTE |
|---|---|---|
| Precision | 0.00 | 0.33 |
| Recall | 0.00 | 0.13 |
| F1-Score | 0.00 | 0.19 |

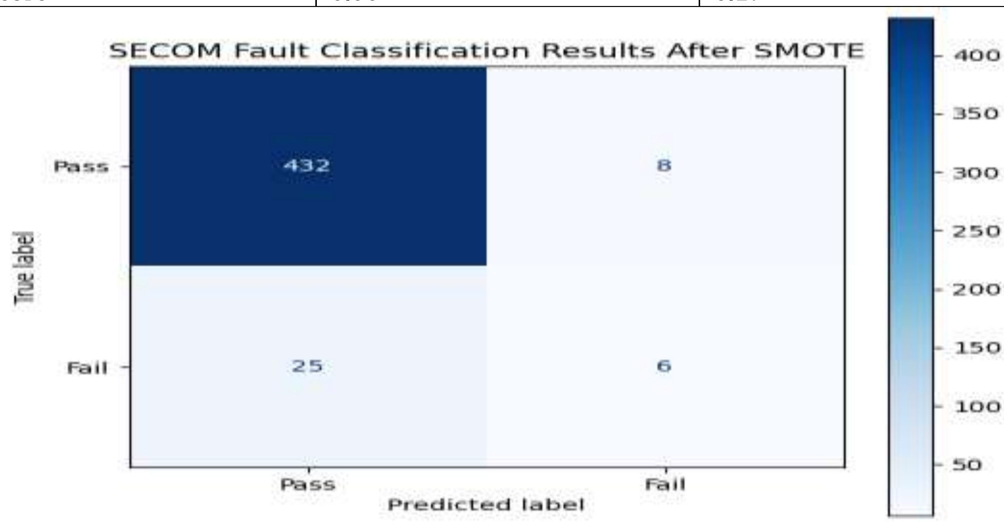


**Figure 5. Confusion Matrix After SMOTE.**

The greatest result is evidence of simulations; recall increased from 0.00 to 0.13. In high-value manufacturing, even detecting a small fraction of defects in advance can reduce rework time and warranty costs. The result represents the validity of simultaneously operating a pipeline with SMOTE-based sampling and PCA for imbalanced, high-dimensional model diagnostics.

### 5.3. Framework Performance and Business Impact

The simulation provided strong quantitative evidence to support the proposed framework's design. The latency simulation (Figure 6) showed that the hybrid-edge architecture delivered 93.28% less decision latency than a cloud-only solution (10.09 ms vs. 150.27 ms), an important component of a responsive Digital Twin.

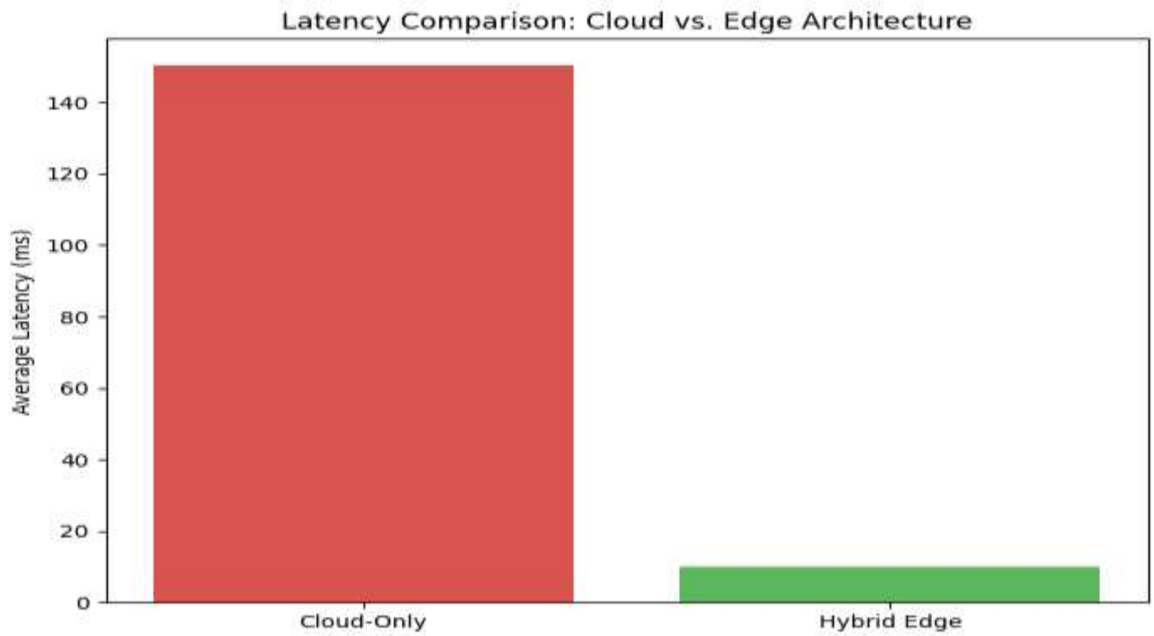


**Figure 6. Latency Comparison: Cloud vs. Edge Architecture.**

The factory simulation quantified the operational value of the framework. The predictive maintenance system transformed total downtime (Table 5 and Figure 7), representing a 75.51% reduction in total downtime.

**Table 5. Simulated Business Impact over a 1-Year Period**

| Metric | Baseline (No PdM) | With PdM Framework | Improvement |
|---|---|---|---|
| Total Downtime | 245 days | 60 days | 75.51% Reduction |
| Total Throughput | 180,050 units | 179,500 units | -0.31% |

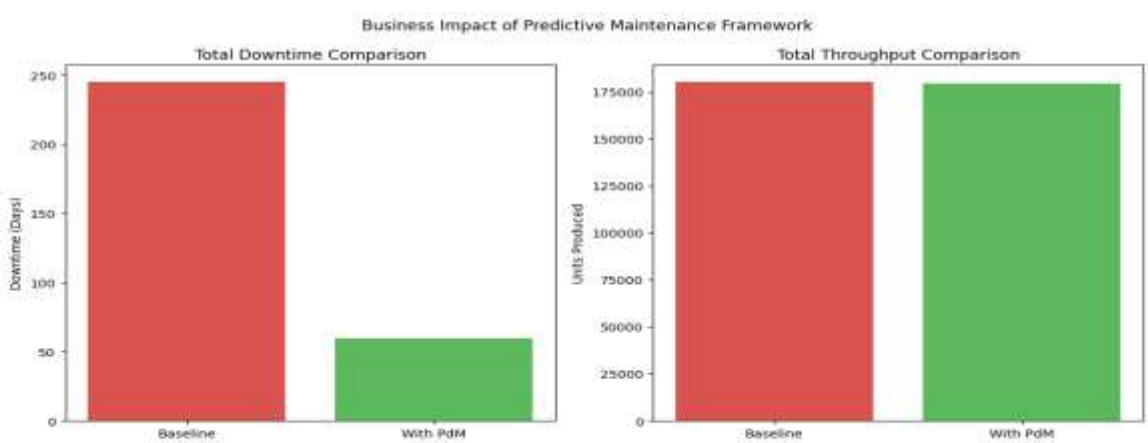


**Figure 7. Business Impact of the Predictive Maintenance Framework.**

The findings present a crucial strategic insight: a negligible decrement to throughput (-0.31%) for increased operational stability and elimination of unplanned down-time was an acceptable trade-off for the gains presented above. As presented in previous analysis, this effectively validates the business case for the framework presented, where the Digital Twin's predictive insights drive more efficient and reliable operations.

## 6. Conclusion

This study has demonstrated the design and implementation of an AI-enabled IIoT framework for predictive maintenance and its validation through a multi-faceted validation methodology. By addressing RUL prediction on time-series data and fault classification on high-dimensional, imbalanced data, the study provided evidence of the versatility and robustness of the framework on different industrial problems. The use of these AI models to act as the analytical engine for a functional Digital Twin, underpinned by a low-latency edge architecture, marks a step towards intelligent, proactive manufacturing systems.

Key Contributions and Findings:

1. A direct evidence-based comparison between an LSTM model and a hyperparameter-tuned XGBoost model for RUL prediction as the context showed that LSTMs outperformed in handling sequential data.
2. A feature importance analysis that provided the RUL model with interpretability and can link the predictions of the model to physically meaningful sensor data, thereby strengthening its perceived value as a component of a Digital Twin.
3. A robust pipeline methodology that employed SMOTE and PCA that provides a real solution to the prevalence of the fault classification problem with imbalanced data that is common in an industrial environment.
4. An evaluation of results of the proposed hybrid architecture, which illustrated the reduction of up to 93.28% latency as a result of edge computing and a simulated business value that revealed a 75.51% reduction in downtime.

### 7. FUTURE SCOPE

This article offers a detailed plan to create predictive maintenance systems. Future research should address the implementation and validation of this framework with actual sensor data on physical edge devices. More research on blockchain for securely logging data may also be done as well as extending the Digital Twin concept to create a more dynamic, physics-based 3D model for higher-level "what-if" scenario analysis.

### ACKNOWLEDGEMENT

The authors would like to thank their respective institutions for providing support during the course of this research.

### DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

Generative AI tools and technologies were used only for language refinement and formatting improvements. The authors are fully responsible for the content.

### FUNDING SUPPORT

The research was self-funded by the authors. No external funding was received for this work.

### AUTHOR CONTRIBUTIONS

Author A developed the concept and prepared the initial draft. Also served as the Corresponding Author responsible for all journal communications; Author B contributed to literature review and content improvement; Author C assisted in methodology and structure; Author D supported editing and formatting; Author E performed major revisions, supervised the work. All authors approved the final manuscript.

**CONFLICT OF INTEREST**

All authors declare that they have no conflict of interest regarding the publication of this paper.

**APPENDIX: Graphical Abstract**

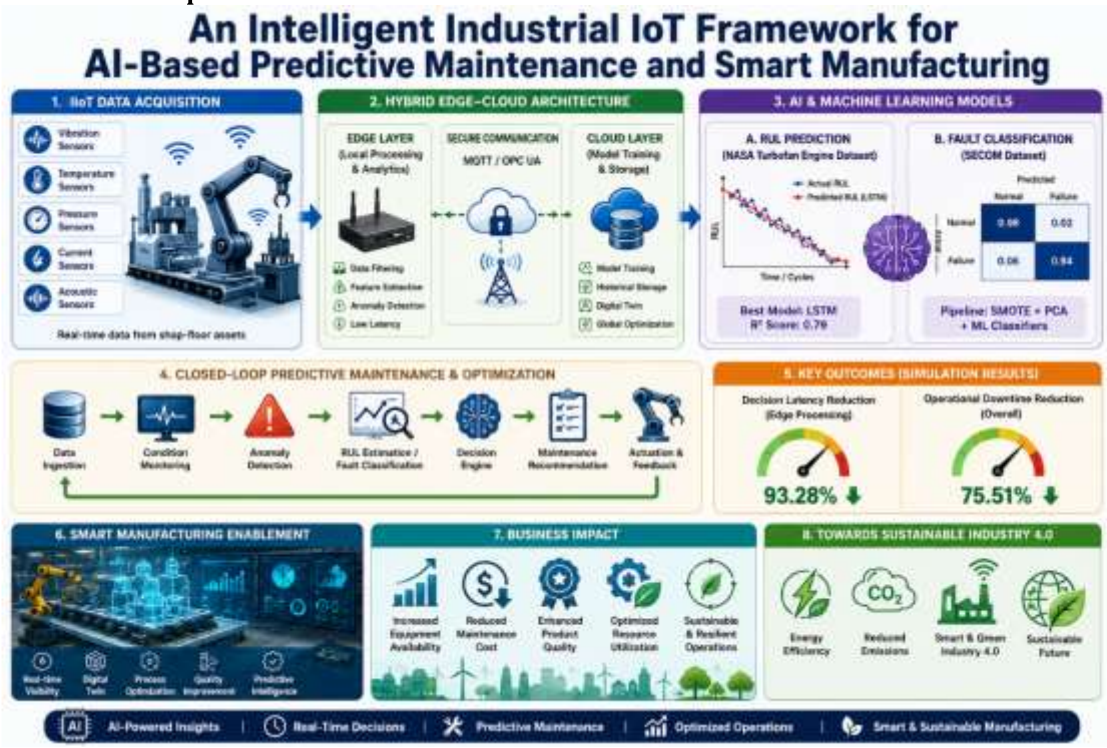


**Figure A1.** Proposed AI-enabled Industrial Internet of Things (IIoT) framework for predictive maintenance and smart manufacturing. The framework integrates real-time industrial sensing, hybrid edge–cloud computing, AI-based Remaining Useful Life (RUL) prediction and fault classification, and a closed-loop decision support mechanism to enable predictive maintenance, operational optimization, reduced latency, and sustainable Industry 4.0 manufacturing

## References

[1] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," *Final report of the Industrie 4.0 Working Group*, Frankfurt, Germany, 2013.

[2] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18-23, Jan. 2015.

[3] Deloitte, "Predictive maintenance and the smart factory," *Deloitte University Press*, 2017.

[4] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483-1510, Oct. 2006.

[5] F. P. Carnero, "Predictive maintenance in the manufacturing sector," *Computers in Industry*, vol. 57, no. 8-9, pp. 664-675, 2006.

[6] F. T. S. Chan, A. Y. L. Chong, and S. H. Choi, "A review of applications of the Internet of Things in manufacturing," *International Journal of Production Research*, vol. 58, no. 8, pp. 2351-2373, 2020.

[7] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, methods and applications," *IEEE Access*, vol. 7, pp. 3343-3354, 2019.

[8] W. He and L. Wang, "A review on big data in manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 171-180, 2018.

[9] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146-157, 2018.
[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016.
[11] A. C. C. de Faria, J. C. C. da Silva, and L. F. M. de Moraes, "Predictive Maintenance in Industry 4.0: A Systematic Literature Review," *Computers & Industrial Engineering*, vol. 150, p. 106889, Dec. 2020.
[12] J. Yuan, H. Liu, and Y. Zhang, "Remaining useful life prediction of aero-engine based on deep long short-term memory network," in *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2016, pp. 1-7.
[13] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, 2019.
[14] J. A. Carino, J. A. Olvera-Guerrero, and E. E. G. Clua, "Predictive maintenance of hydraulic systems using XGBoost," in *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, 2018, pp. 1-6.
[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.
[16] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009.
[17] K. Kerdprasop and N. Kerdprasop, "An analysis of machine learning techniques for semiconductor manufacturing classification," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2007.
[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.
[19] G. Haixiang et al., "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220-239, 2017.
[20] H. J. K. Lee and J. H. Lee, "Process monitoring using principal component analysis," *Chemical Engineering Science*, vol. 55, no. 2, pp. 223-233, 2000.
[21] S. Mocnej, S. Kavacky, and I. Zolotova, "MQTT-based communication for IoT," in *2017 18th International Carpathian Control Conference (ICCC)*, 2017, pp. 289-294.
[22] M. Syafrudin, G. Alfian, F. Fitriyani, and J. Rhee, "A holistic approach for anomaly detection in sensor data for smart manufacturing," *Sensors*, vol. 18, no. 12, p. 4344, 2018.
[23] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219-235, Aug. 2019.
[24] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry 4.0: A review of recent research and applications," *International Journal of Production Research*, vol. 57, no. 12, pp. 3543-3567, 2019.
[25] S. Wang, J. Wang, X. Wang, T. Qiu, and Y. Yuan, "A survey on deep learning for anomaly detection in industrial internet of things," *IEEE Access*, vol. 8, pp. 119565-119580, 2020.
[26] B. 3D, "NASA CMAPS dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/behrad3d/nasa-cmaps.
[27] M. Lichman, "UCI Machine Learning Repository," Irvine, CA: University of California, School of Information and Computer Science, 2013. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SECOM.