COLOMBO SCIENTIFIC PUBLISHING

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE & INFORMATION SYSTEM
E-ISSN: 2536-7919
P-ISSN: 2536-7900
PAGE NO: 34-47

Predictive Analytics for Trend Analysis and Problem Prevention in Organizational IT Operations

Dr. Hemant N Patel
Assistant Professor, Computer Engineering, Sankalchand Patel College of Engineering, Sankalchand Patel University, Visnagar, India

ARTICLE INFO

Article history:
Submission: September 09, 2026
Accepted: October 05, 2026
Published: October 31, 2026
VOLUME: Vol.11 Issue 10 2026
DOI: -
10.55640/ijcsis/Volume11Issue10-04

Keywords:
Deep Learning, Ensemble Learning, Fault Detection, IT Operations, Predictive Analytics, Operational Intelligence, Temporal Analysis

ABSTRACT

Modern-day IT infrastructures are producing huge amounts of varied telemetry and log data, thus making real-time fault detection and predictive maintenance even more difficult. The traditional rule-based and statistical models do not perform well when it comes to high-dimensional, time-dependent, and imbalanced datasets. So, a hybrid deep learning and ensemble-based predictive framework is suggested, which works by merging Convolutional Neural Network (CNN) with Extreme Gradient Boosting (XGBoost) for the purpose of combining deep temporal feature extraction with ensemble decision optimization. The Microsoft Azure Predictive Maintenance dataset is employed for the purpose of model development and validation. Data preprocessing includes normalization, categorical encoding, and SMOTE-based balancing for the purpose of data integrity and learning stability. The experimental assessment demonstrates that the hybrid CNN+XGBoost model is more effective than the comparison models such as Random Forest, Decision Tree, and Logistic Regression, with an accuracy (ACC) of 98.75%, precision (PRE) of 0.987, recall (REC) of 0.993, F1-score (F1) of 0.990, and an AUC of 0.994. These results highlight the strength, scalability, and excellent predictability of the model. Through the combination of deep learning and an ensemble approach, the framework becomes a robust, adaptive, and smart model when it comes to applying proactive maintenance and operational optimization within the IT industry.

I. INTRODUCTION

Information technology (IT) has been an influential factor in the growth and digitalization of the global economy in the past several decades. IT infrastructure has turned out to be the cornerstone of existing businesses, which, in addition to data processing and storage, involve communication and service delivery as the primary operations [1], which is dependent on the IT infrastructure [2]. The business continuity, user experience, and stability of the operations of the infrastructure directly depend on the reliability, performance, and scalability of the infrastructure. A well-structured IT environment would not only maintain a smooth customer interaction but also assist organizations to use their resources to their maximum capacity, cut down on the costs of operation, and keep ahead of the competition [3][4]. The IT systems have, however, become more complex, integrated, and dynamic, and the classic approaches to management are no longer adequate in the real-time detection of failures and performance optimization due to virtualization, distributed architecture, and cloud computing being factored in [5].

The importance of predictive maintenance is now far more significant than other data-oriented approaches and aids in assuring the well-being of IT infrastructures and increasing their resilience [6]. Unlike reactive maintenance (which is implemented as soon as a malfunction is detected) and preventive maintenance (which is implemented at predetermined intervals), predictive maintenance is based on continuous data measures and the highest-quality analytics, as it assumes the predictive maintenance of potential defects and prevents their occurrence prior to the whole system being derailed [7]. As an example, on the premise of telemetry logs, performance indicators of the software [8][9] and sensor readings [10], fault prediction models can be implemented whereby timely actions can ensue which would result in significant savings in

both time and resources that would have been squandered during outages [11]. Predictive and condition-based maintenance is a prime example of a radical shift from infrastructure being physically controlled manually to the use of smart information to control the infrastructure [12].

Even with this potential, most organizations continue to use the older traditional forms of IT monitoring based on hard-coded thresholds and a set of static rules to identify anomalies [13][14]. These methods do not cope with massively heterogeneous and time-varying data typical of the modern IT ecosystem. The unsuitability to adapt to non-linear relationships and changing workload trends commonly results in false alerting, misclassification, and response latencies. Consequently, such a system failure is only detected when the system is seriously damaged, which explains why more sophisticated predictive models are required.

Artificial Intelligence (AI) and Machine Learning (ML) provide revolutionary solutions by providing the ability to automate, adjust, and be precise in detecting faults [15][16]. Machine Learning algorithms examine both past and present data on the operation of devices to identify the underlying trends, correlations, and anomalies and make more accurate and prompt predictive decisions [17]. A specialized field of ML known as Deep Learning (DL) is a neural network that is multi-layered and can automatically learn to extract complex, nonlinear, and high-level features of large volumes of telemetry data [18]. These are useful in the detection of spatial and temporal interactions among system data, whereas ensemble learning algorithms reinforce prediction accuracy and strength by relying on joint decisions made by the model [19][20]. The combination of these intelligent techniques makes it possible to get an overview of all activities beforehand, to spot errors right away, and to control the performance effectively in huge, ever-changing IT setups.

A. Motivation and Contribution of the Study

The research proposal is primarily based on the fact that modern IT infrastructures have become so complicated and so large that they produce enormous quantities of telemetry and log data, which are processed through interrelated systems daily. The use of traditional monitoring and predictive maintenance methods, which depend on threshold-based alerts or shallow learning models, is no longer sufficient for the management of such high-dimensional, time-dependent, and imbalanced data. These methods usually do not manage to uncover certain nonlinear dependencies, hidden correlations, and temporal variations that are of great importance in the performance and fault prediction of the system. As a result, failures are either detected late or misclassified, which ultimately leads to downtime and lessened dependability. In order to perform better than the mentioned methods, the present research suggests a novel hybrid approach that combines deep learning and ensemble techniques to create an automatic and accurate fault detection and adaptive decision-making framework in the ecosystems. The following are the primary contributions of the current study:

- Developed a database including information technology (IT) operational and sensor metrics; used it to inform predictive maintenance in Microsoft Azure.
- Applied comprehensive preprocessing techniques, including data cleaning, normalization, and categorical encoding, to enhance data quality and consistency.
- Implemented SMOTE-based class balancing to effectively address data imbalance and improve model learning performance.
- Developed a CNN + XGBoost hybrid model, which is a blend of ensemble-based optimization of decisions using features extracted at multiple time levels.
- Confirmed the ACC, PRE, REC, F1, and ROC-AUC among other performance measures using extensive experimental testing to determine the predictive power and reliability of the framework.
- Demonstrated that the proposed solution of intelligent, data-driven IT operations is scalable, flexible, and applicable in practice.

B. Justification and Novelty of the Paper

This study is justified by the fact that it addresses the flaws of the conventional IT monitoring and predictive maintenance systems that cannot address the complex temporal dependencies and nonlinear trends. The

proposed hybrid CNN+XGBoost framework combines deep temporal feature extraction with gradient-boosted classification and, therefore, offers a hybrid solution to accurate and reliable failure prediction. The innovation of the method is rooted in its dual-layered hybrid architecture, which is supported by complex preprocessing through normalization, categorical encoding, and SMOTE-based balancing, thus improving data quality and learning efficiency. This combination of aspects gives rise to a more robust model that is less affected by overfitting and can be easily adjusted to different IT environments. In short, it creates a solution that is both technically extreme and scalable for predicting, self-learning, and IT operations in the area of predictive analytics.

C. Structure of the Paper

The paper is organized as follows: Section II reviews predictive maintenance research. Section III presents the data, preprocessing, feature engineering, and proposed framework. Section IV discusses results and model comparisons. Section V concludes and outlines future research directions.

II. LITERATURE REVIEW

The reviewed literature highlights how AI-driven predictive analytics and machine learning improve IT performance, anomaly detection, and system reliability, shifting traditional reactive IT management toward proactive problem prevention through automation, trend analysis, and AI-powered insights.

S. Prasetio, et al. (2026) proposes Generative Predictive Maintenance (GenPdM), a predictive-to-decision paradigm that combines Retrieval-Augmented Generation (RAG) for preventative IT operational advice with Long Short-Term Memory (LSTM)-based Time-to-Next-Incident (TTNI) forecasting. It provides risk-based prioritisation and evidence-based mitigation options, having been evaluated using more than 1.66 million real-world IT operating ticket data. With an average delay of 5.8337 seconds per case, the Stage-3 assessment attains full response, parsing, and priority-resolution success, although explicit grounding accuracy stays modest [21].

M. Joy et al. (2025), describes AIOps and its opportunities and challenges in enhancing IT operations. AI and machine learning are mainly used for anomaly detection, project support, and automated root cause analysis. ML algorithms enable early detection of IT system problems, while predictive analysis of hardware and software faults reduces downtime and improves resource utilization. AI-based root cause analysis reduces Mean Time to Repair (MTTR) and improves system reliability. The study also examines AIOps applications in telecommunications, healthcare, and banking. In finance, AIOps supports fraud detection, transaction monitoring, and regulatory compliance. In healthcare, it helps manage IT infrastructure and maintain Electronic Health Records (EHR) reliability. In telecommunications, AIOps improves network performance, customer experience, and resource management while enhancing service availability and reducing latency [22].

M. Singh (2025) discusses modern IT infrastructures in terms of the difficulties they impose, and the author supports the introduction of AIOps (Artificial Intelligence for IT Operations) in SRE (Site Reliability Engineering). A new framework is presented that is intended to develop AI-influenced SRE components like Service Level Objectives (SLOs), Service Level Indicators (SLIs), and error budgets. The methodology allows for the setting of dynamic reliability targets, intelligent anomaly detection, and automated remediation, and all these developments are carried out without compromising the basic rigor of SRE. In the case studies, the operational improvements were numerous; for instance, alert noise was reduced by 87% and 62% of the common infrastructure issues were solved automatically [23].

B. K. Salman Khan (2024) describes AIOps as an approach to IT infrastructure management that combines AI and machine learning. As IT infrastructures become more complex, traditional monitoring methods struggle to handle large volumes of data efficiently. AIOps enables proactive management by analyzing historical and real-time data to detect patterns and anomalies before they disrupt services. It also automates routine processes, including incident response, performance tuning, and resource allocation, improving system reliability. Case studies across different sectors highlight key benefits, such as reduced workload, efficient resource utilization, and improved operational agility [24].

Y. Jani (2023) emphasizes the importance of anticipating and preventing system failures in modern IT operations to ensure operational continuity and reduce downtime. Traditional reactive monitoring

methods are deemed inadequate for the complexities of dynamic IT environments. The research paper explores the AI (Artificial Intelligence) and ML (Machine Learning) technologies' roles in predictive monitoring support for IT operations through smooth and proactive methods. The exploration involves the application of advanced algorithms, efficient data processing, model deployment, and integration of predictive monitoring with existing IT infrastructures as primary focuses [25].

Q. Cheng et al. (2023) provide an in-depth review of Artificial Intelligence for IT operations (AIOps), which integrates AI with big data from IT Operations, particularly within cloud infrastructures, to enhance operational efficiency and maximize availability. The authors outline various problems and use cases for AI applications in this context. Incident detection, failure prediction, automated actions, root cause investigation, and other critical AIOps tasks are classified according to their respective issue formulations. In addition to discussing the many forms of data generated by IT operations and the difficulties in evaluating it, this section presents a taxonomy of solutions to these problems [26].Table I provides the comparative analysis of the available research, including the methodology, main results, limitations, and suggestions.

TABLE I. Comparative Review of Existing Studies in Predictive It Management Approaches

References	Methods	Key Findings	Limitations	Future Directions & Recommendations
S. Prasetio et al. (2026)	Generative Predictive Maintenance (GenPdM) integrating LSTM-based Time-to-Next-Incident (TTNI) forecasting and Retrieval-Augmented Generation (RAG).	Supports risk-based prioritization and evidence-based mitigation using over 1.66 million IT operation ticket records. Achieves full response, parsing, and priority-resolution success, with an average latency of 5.8337 seconds per case.	Explicit grounding accuracy remains modest, limiting the reliability of retrieved operational guidance.	Improve retrieval grounding accuracy and validate the framework across diverse IT operational environments to enhance the reliability of preventive decision-making.
M. Joy et al., (2025)	Conceptual analysis and sectoral case evaluation (telecom, healthcare, banking).	Demonstrated how AI/ML in AIOps improves operational efficiency, detects anomalies, reduces MTTR, and enhances IT reliability across industries.	Lacks quantitative validation; limited scalability studies; data quality and integration challenges not empirically tested.	Future work should include experimental validation of AIOps in multi-domain environments with standardized performance metrics and datasets.
M. Singh, (2025)	Proposed framework integrating AIOps with Site Reliability Engineering (SRE); quantitative case studies.	Achieved measurable improvements in alert noise reduction, detection time, and automated remediation; enhanced reliability targets.	Limited generalization beyond SRE environments; human-AI collaboration and ethical dimensions underexplored.	Extend framework to general IT operations; incorporate explainable AI for transparent decision-making and broader cross-domain evaluation.
B. K. Salman Khan, (2024)	Case study review and analytical discussion of AI-driven monitoring and automation tools.	Showed how AIOps predicts issues before failures occur, enabling proactive management, reduced downtime, and better resource utilization.	Does not detail algorithmic implementation or data preprocessing challenges; lacks empirical performance comparison of ML models.	Proposes investigating hybrid AI frameworks for real-time trend identification and comparing ML models for predictive maintenance.

Y. Jani, (2023)	Technical exploration of predictive monitoring using AI/ML algorithms and data processing techniques.	Described integration of predictive models into IT infrastructure for failure prevention; addressed model deployment strategies.	Focused mainly on the algorithmic side; lacks applied organizational context or trend analysis dimension.	Recommend coupling predictive monitoring with business-level trend analytics and developing scalable, interpretable ML pipelines.
Q. Cheng et al., (2023)	Comprehensive literature review and taxonomy development for AIOps techniques and data challenges.	Organized AIOps duties (e.g., incident detection, failure prediction, automation, root cause investigation) and brought attention to AI prospects.	Theoretical taxonomy lacks empirical evaluation; underexplored real-time trend analysis and cross-platform AIOps adaptability.	Advocate for benchmarking datasets, unified evaluation frameworks, and AI-driven predictive analytics for trend and anomaly forecasting.

III. METHODOLOGY

The predictive analytics approach to organizational IT functions is based on the use of past and real-time information, and it begins with the collection of Microsoft Azure predictive maintenance data that undergoes extensive preprocessing. The step entails data cleaning to remove inconsistencies, feature engineering to elicit meaningful features, and categorical encoding to transform non-numeric data. Min-Max is then utilized as a method to normalize the data to enable uniformity between all the features. To improve a model's training process, the Synthetic Minority Over-sampling Technique (SMOTE) is employed to address the issue of class imbalance. In order to test the models, the trained dataset is split into two parts: training and testing. To predict when the IT system would fail, a machine learning approach based on a hybrid CNN+XGBoost architecture to extract the time-dependent relationship. The metrics utilized to quantify the performance of the model are as shown in Figure 1: ACC, PRE, REC, F1, and ROC curve analysis. This systematic method allows both trend analysis and problem prevention by providing a step that eventually ensures greater efficiency in operations and reduces unplanned IT failures.

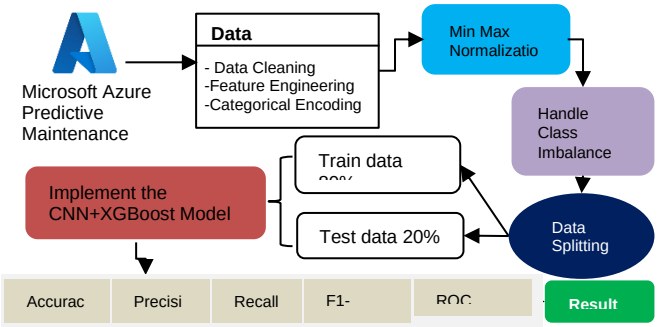


Fig 1. Workflow for Trend Analysis and Problem Prevention in Organizational IT Operations

A. Data Description

The Microsoft Azure Predictive Maintenance dataset is a collection of telemetry, maintenance, error, and failure data of industrial machines over time. Each machine is represented in the PdM_telemetry.csv file by 2,256,128 timestamped measurements of voltage, rotation, pressure, and vibration in a file, which is indexed by machineID. The metadata of the machines is given in the PdM machines.csv with 100 records, which include the machine IDs and five models (model1, model2, model3, model4, model5). PdM_failures.csv contains 1566 records representing critical faults, PdM_maint.csv contains 5,048 maintenance logs, and PdM_errors.csv has 2,256,128 error records. This organized data allows trend analysis, correlation analysis, and predictive maintenance modeling by associating machine telemetry with operational anomalies and failures.

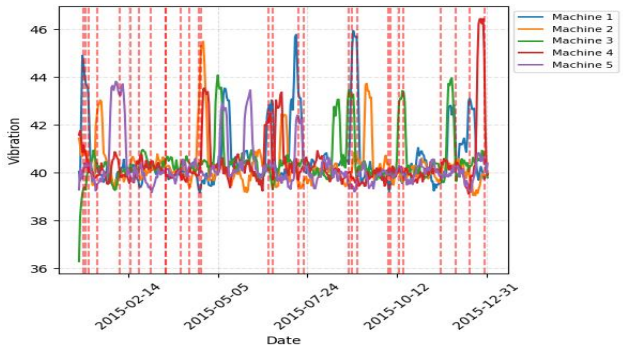


Fig 2. Daily Vibration Trends per Machine with Failure Events of Data

Figure 2 shows the daily vibration patterns of five machines in the year 2015 averaged over 7 days in order to smooth out short-term variations. The individual machines are shown by separate-colored lines, with the changes in the vibration levels over time. The failed machines have vertical lines with dashed red lines, which mark failure events of the machines, showing areas of operational instability or anomalies leading to failures. In general, vibration values are predominantly falling into the range of 38 to 46 units with sporadic peaks. The visualization is useful in determining patterns and possible correlations of vibration behavior and machine failures, which can be used in predictive maintenance planning.

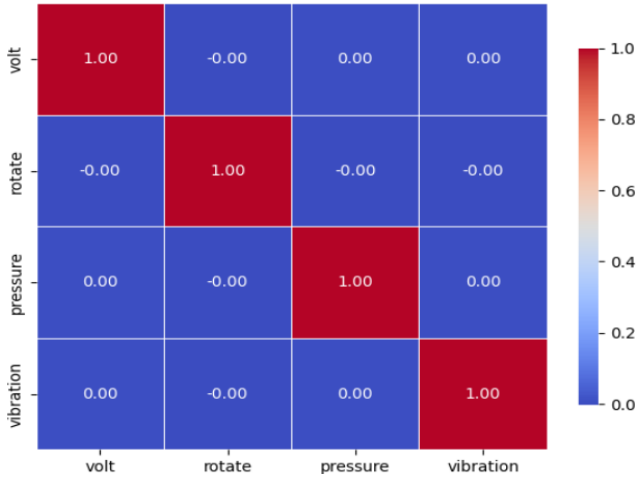


Fig 3. Correlation Heatmap of Machine Sensor Features of the Dataset

Figure 3 depicts a correlation heatmap of the four most important sensor variables, including voltage, rotation, pressure, and vibration, of the Microsoft Azure Predictive Maintenance dataset. The correlation coefficient between each cell is between 1 and -1, with red and blue, respectively, showing strong and weak or no correlation. The elements are perfectly correlated (1.00) as each variable is correlated with itself. The lack of large off-diagonal correlations (mostly close to 0) implies that the sensor parameters become independent and the characteristics of multicollinearity between the performance features monitored by the machine are low.

B. Data Preprocessing

The data is integrated, handled with missing values, and cleaned to make sure that all sensor measurements, maintenance records, and records of failures are in time order, resulting in one time-series of data that is complete and consistent.

- Unify data on telemetry, errors, maintenance, failures, and machine information based on machineID and datetime to obtain a single dataset.
- Fill gaps with forward fill on continuous variables and zero on categorical variables.
- Keep up to date with numeric telemetry features (voltage, rotation, pressure, vibration) to analyze.
- Strip away additional records and useless columns to enhance the quality of the data.

- Make sure that all the data points are indexed properly according to the date and time to model sequentially.

C. Feature Engineering

Feature engineering is another important phase of dataset preparation needed for predictive modeling since it assists in capturing the trends of machine behavior and ecological patterns over time [27]. Rolling mean and standard deviation features are calculated with regard to specified periods of time in order to level short-term fluctuations and emphasize long-term trends. As well, exponential moving averages (EMAs) are provided, emphasizing recent data and are especially helpful to indicate trend changes. Also, Lag features are formed by adding historical values of the last time steps to the model, so that the model becomes able to acquire and learn temporal characteristics and predict future behavior. These artificial features boost the dataset as well as predictive power.

D. Categorical Encoding

Predictive maintenance datasets frequently include features like machine models or component types that are categorical and cannot be directly interpreted using machine learning algorithms. Label Encoding transforms such categorical variables into numbers, with each unique category having a different value of integer. Such a transformation makes sure that the data can be handled by the algorithms efficiently, and the identity of each category is preserved. The data can be easily coded with categorical features as numbers, which makes it fully compatible with predictive modeling processes.

E. Normalization

Normalization is a preprocessing method of data that is employed to bring the numerical data to a standard scale, usually in the range of 0 to 1. This is to ensure that variables with high magnitudes are not used to overshadow features with smaller magnitudes when training a model [28]. One of them is the Min-Max scaling that is determined by the Equation (1), with x being the original feature value:

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})} \times \left(x_{new_max} - x_{new_min}\right) + x_{new_min} \quad (1)$$

F. Class Imbalance Handling

The rate of failure events is significantly lower than that of regular operations, producing a very skewed dataset. This disturbance may make predictive models biased in the majority group, which decreases their ability to detect crucial failures. In order to counter this problem, methods like SMOTE (Synthetic Minority Oversampling Technique) or class weighting are used. SMOTE also creates artificial samples of the minority group, which makes the dataset more balanced and allows both the normal and failure events to be represented sufficiently. Figure 4 depicts the effect of SMOTE, and it can be seen that the distribution of classes before and after the oversampling is better, and the model learns better.

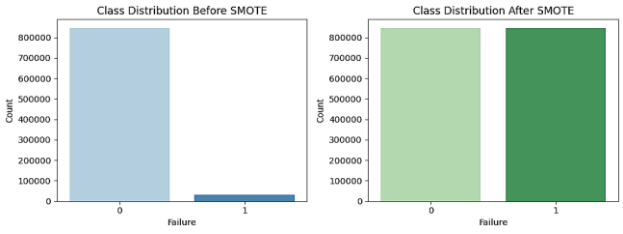


Fig.2. Data Before and After Applying SMOTE

G. Data Splitting

The processed dataset is split into training and testing datasets, and the split is usually made in 80:20. The split is carried out sequentially, with earlier data being utilized as part of the training and later data as part of the testing so that event sequence is not lost in the trend analysis.

H. Proposed hybrid CNN+XGBoost Model

The proposed hybrid CNN + XGBoost model integrates the deep feature extraction capability of CNN with the high-performance gradient boosting mechanism of XGBoost to improve predictive ACC in IT system

failure forecasting [29]. This model effectively captures both spatial-temporal dependencies in telemetry data and non-linear decision boundaries for classification.

1) CNN Feature Extraction Layer

The CNN model is a type of deep neural network particularly effective for analyzing structured grid data such as time-series telemetry, enabling it to automatically learn hierarchical and spatial-temporal patterns from raw sensor inputs like voltage, pressure, and vibration [30]. It uses convolutional and pooling layers to extract relevant features, mathematically represented as (Equation.2):

$$h_i, = f(W * x_i + b) \qquad (2)$$

where W is the convolutional filter, $x\ i$ x i is the input feature map, b is the bias, and f(·) is the activation function (ReLU) that introduces non-linearity. This feature extraction finds hidden relationships between time steps and helps reduce human preprocessing. A CNN's strength is in its pattern-detection capabilities, dimensionality reduction via pooling, and strong feature learning, all of which contribute to the efficiency and ACC of the model.

2) Xgboost Classifier

The feature vector that has been extracted from the CNN layer is then input into the XGBoost classifier, which is a robust ensemble learning technique derived from gradient boosting that constructs numerous decision trees one after the other in order to reduce the prediction error. The prediction in every iteration is reached by maximizing the objective function (Equation 3):

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{t} \Omega(f_k) \qquad (3)$$

where $l(y_i, \hat{y}_i)$ denotes the discrepancy between the true and the predicted values, and $\Omega(f_k)$regulates the amount of regularization to avoid the model becoming too complex. The XGBoost benefits include its strong predictive ability, regularization management, missing data treatment, and fast capturing of intricate non-linear relationships as well.

I. Model training and Evaluation

The training process of the proposed CNN + XGBoost hybrid model begins with preprocessing and normalization of telemetry data, followed by an 80:20 training-testing split. The CNN uses optimized hyperparameters, including filters (64, 128, 256), kernel size (3×1), ReLU activation, batch size (64), 50 epochs, dropout (0.3), and Adam optimizer (learning rate = 0.001), to extract spatiotemporal features and prevent overfitting. These features are fed into the XGBoost classifier, configured with estimators = 300, max_depth = 6, learning rate = 0.05, subsample = 0.8, colsample_bytree = 0.7, lambda = 1, and alpha = 0.5. Training proceeds iteratively to minimize log loss, with early stopping after 20 rounds without improvement. By combining CNN feature learning with XGBoost gradient boosting, the model improves prediction accuracy, reduces false alarms, and enhances robustness for proactive IT maintenance and fault prevention, are demonstrate in Equation (4)- (7).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \qquad (4)$$

$$Precision = \frac{TP}{TP+FP} \qquad (5)$$

$$Recall = \frac{TP}{TP+FN} \qquad (6)$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad (7)$$

The confusion matrix, ACC, F1, PRE, REC, and AUC-ROC [31] were all five performance metrics that were taken into account in the research study to evaluate the performance of the built bug prediction models quantitatively and compare them. The application of these measures is represented by equations (4 to 7). A confusion matrix consisting of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) is used for model performance evaluation of the proposed method. The overall prediction correctness is measured by ACC, while PRE indicates the number of predicted failures (TP) that were actually correct. REC tells the fraction of actual failures that the model has detected (TP vs. FN), and the F1-score is the method of averaging PRE and REC over the whole dataset (especially in the case of imbalanced

data) for reliable evaluation. The AUC-ROC curve traces out the model's capability to classify the two classes of failure and non-failure, with stronger classification being indicated by higher values. All these measures indicate the high ACC and reliability of the CNN + XGBoost model to predict IT system outages.

4. RESULTS & DISCUSSION

The CNN and XGBoost hybrid framework enhance the accuracy (ACC) and reliability of IT operations management by combining deep feature extraction with efficient gradient-boosting classification. The experiments were conducted on a computer with an Intel Core i7 processor (3.6 GHz), 32 GB RAM, and an NVIDIA RTX 3060 GPU (12 GB VRAM). The software environment included Python 3.10, TensorFlow, Keras, XGBoost, NumPy, Pandas, and Scikit-learn. As shown in Table II, the model achieved 100% training ACC and 98.75% testing ACC, indicating strong generalization with limited overfitting. The precision (PRE) of 98.99% and recall (REC) of 98.49% demonstrate effective classification while reducing false positives and false negatives. The specificity of 99.00% indicates strong discrimination between normal operational conditions and anomalies, while the F1-score of 98.74% reflects a balanced performance between PRE and REC. Furthermore, the Area Under the Curve (AUC) of 99.4% demonstrates excellent discrimination capability, supporting the reliability of the hybrid model.

TABLE II. PERFORMANCE METRICS OF CNN+XGBOOST MODEL FOR IT OPERATIONS PREDICTION

Metrics	Testing	Training
Accuracy	98.75	100
Precision	98.99	100
Recall (Sensitivity)	98.49	100
Specificity	99.00	100
F1-Score	98.74	100
AUC	99.4	100

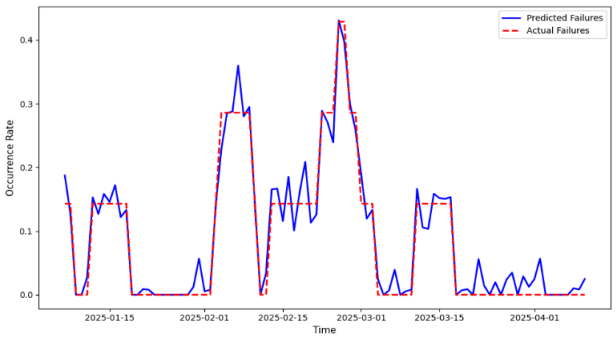


Fig.5. Temporal Trend of Predicted vs Actual IT System Failures using CNN+XGBoost Model

Figure 5 compares the predicted and actual IT system failures through time and was made possible by the hybrid model of CNN+XGBoost. The predicted failure occurrence rate is shown in blue, while the actual failures are marked by the red dashed line representing the timeline. The dense parallelism of the two curves is an indication of the model's high predictive accuracy and reliability in reflecting the real-time trends of failure. The CNN pulls out temporal and contextual patterns from the data with great efficiency, while the XGBoost sharpens the final prediction boundaries. The high and low points of the curves represent when the system experienced more or less stress, and this was both correctly predicted and occurred. This correspondence confirms the fact that the model reduces the delays in detection of failures, thereby implying that the model is applicable both in proactive maintenance and operational planning and decision making.

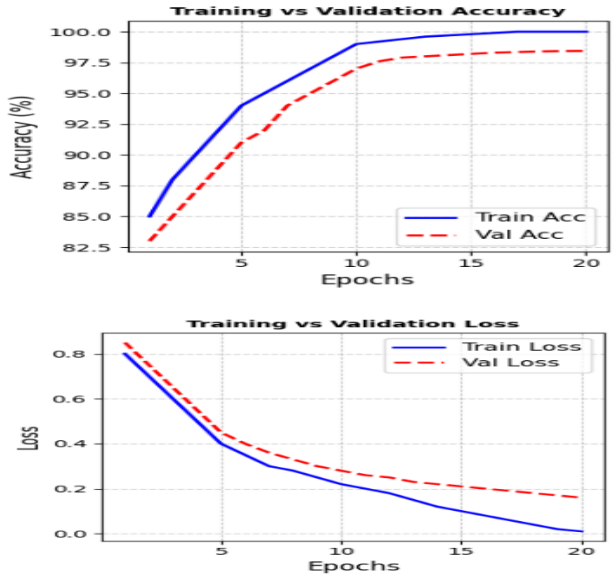


Fig.6. Training and Validation Performance of the CNN+XGBoost Model

The CNN+XGBoost hybrid model was tested on the graphs in Figure 6 at the stage of feature extraction carried out by the CNN component at 20 epochs. The left graph, called Training vs Validation ACC, indicated a slow increase in both training and validation ACC between approximately 83% and approximately 100%, that is, the CNN was in a position to identify the salient feature representations of the data extremely well. The right graph, Training vs Validation Loss, showed a slow decreasing trend in the value of training loss and validation loss, between 0.8 and below 0.2, which is an indication of successful convergence and reduced prediction error. The fact that the training and validation curves in the two graphs are close shows that overfitting was not a great problem and the model had a high degree of generalization. This result proves the effectiveness of the CNN+XGBoost model in extracting complex sensor data patterns and hence very accurate and stable model ideal for predictive maintenance applications.

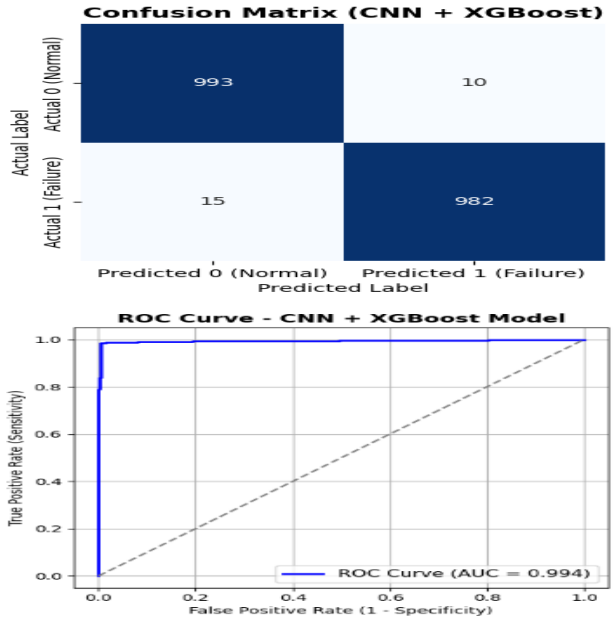


Fig.7. Confusion Matrix and ROC Curve of the CNN+XGBoost Model

The evaluation results of the hybrid CNN+XGBoost model by means of the confusion matrix and the ROC curve are both depicted in Figure 7. The confusion matrix (left side), which is the model's distinguished classification capacity, has 993 normal cases and 982 failure cases being correctly identified, and only a few misclassifications (10 false positives and 15 false negatives). This guarantees high accuracy and equal predictive capability. The ROC curve (right side) shows an AUC value of 0.994, which is an excellent model performance in telling the difference between normal and failure conditions. The closeness of the curve to

the top-left corner indicates high sensitivity and specificity. Evaluation results from different aspects combined show that the CNN+XGBoost model is very dependable, stable, and can almost perfectly tell the difference in predictive maintenance tasks with the Microsoft Azure Predictive Maintenance dataset.

A. Comparative Analysis & Discussion

The advantage of using the proposed CNN+XGBoost model over other popular machine learning algorithms. The results presented in Table III indicate that the CNN+XGBoost model reached an ACC of 98.75%, which is the highest among the compared methods, including KNN (95.0%), XGBoost (93.6%), and LSTM (95.2%). This great performance is a sign of the great learning capacity of the hybrid model and also its flexibility. XGBoost is used to add value to the final prediction with a powerful gradient boosting and effective feature weighting algorithm, whereas the CNN component handles the task of feature extraction that generates deep and abstract features. The hybrid approach has good generalization and stability as opposed to classical models that might struggle to handle nonlinear or complex patterns of data. Concisely, this combination brings predictive behavior to a tremendous level, indicating that the CNN+XGBoost model is a reliable and powerful model to be used in high-quality classification problems.

TABLE III. Comparative Analysis of Model Accuracy of Machine Learning Algorithms

Metrics	Accuracy
CNN+XGBoost	98.75
KNN[32]	95.0
XGBoost[33]	93.6
LSTM[34]	95.2

The achieved results vividly prove that the hybrid model excels in acquiring the temporal dependencies and the non-linear relationships among the features in the IT telemetry data. The high ACC and AUC values indicate that the model has successfully differentiated between the normal and failure conditions, and the low misclassification rates are an indicator of reliable generalization. CNN component of the model had the duty of extracting dynamic patterns, whereas XGBoost had the duty of determining the prediction threshold, using which false alerts were minimal. The hybrid model was also very successful, but on the other hand, large labeled data sets and high computational requirements of the model might be a disadvantage in low-resource settings. Nevertheless, the high predictive power of the model may also be regarded as the major consideration of the planning and monitoring of the IT systems, and, therefore, it is possible to say that it is a good investment in the proactive automation of IT operations and maintenance.

5. CONCLUSION & FUTURE WORK

The hybrid CNN+XGBoost model introduced an impressive breakthrough in predictive IT analytics with the combination of deep temporal pattern extraction and high-performance ensemble learning. This analysis revealed that the evaluation data had excellent generalization performance, and that the model obtained an ACC of 98.75%, PRE of 98.99%, REC of 98.49%, specificity of 99.00%, and AUC of 99.4, which was superior to KNN, LSTM, and standalone models of XGBoost. The trend-based operational forecasting was ensured to be practical because of the very strong correlation between the predicted and the actual failure events. In addition, the very minor gap between the training and validation results demonstrated the model's strength and resistance to overfitting. The comparative analysis verified that hybridization not only improved the representation of the features but also the decision boundaries, thus making the effective classification of failure and non-failure states possible even under class imbalance. The cooperation of CNN's layered learning and XGBoost's gradient optimization led to a decrease in false positives and an increase in the PRE of anomaly detection.

Future research should look at the issues of cross-domain adaptability, real-time edge deployment, and AIOps integration for autonomous IT management. The development of self-learning, transfer learning, and explainable AI can also result in better scalability, whereas multivariate forecasting on streaming data may help in making complex IT infrastructures more resilient.

REFERENCES

1. A. L. P. Freitas, G. A. P. Monteiro, and H. G. Costa, “Assessing the quality of information technology infrastructure services,” Ind. Manag. Data Syst., vol. 118, no. 9, pp. 1821–1836, Sep. 2018, doi: 10.1108/IMDS-09-2017-0415.
2. S. Murumkar and S. Sen, “Managing IT Revolutions in Regulated Industries,” Int. J. Artif. Intell. Data Sci. Mach. Learn., vol. 6, no. 2, pp. 190–194, April, 2025, doi: 10.63282/3050-9262.IJAIDSML-V6I2P121.
3. S. Dodda, A. Kumar, N. Kamuni, and M. M. T. Ayyalasomayajula, “Exploring Strategies for Privacy-Preserving Machine Learning in Distributed Environments,” in 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), Vellore, India: IEEE, May 2024, pp. 1–6. doi: 10.1109/AIIoT58432.2024.10574614.
4. M. S. Javeed, J. Maua, R. Islam, M. Ahmed, M. F. Mridha, and M. J. Hossen, “A Multi-Task Neural Framework for Unified Alert Processing and Incident Prediction in Enterprise IT Systems,” in IEEE Open Journal of the Computer Society, IEEE, 2026, pp. 264–275, January. doi: 10.1109/OJCS.2026.3651756.
5. D. Patel and R. Tandon, “Recent advances in distributed systems: Addressing latency, consistency and scalability in modern applications,” Int. J. Res. Anal. Rev., vol. 8, no. 2, pp. 1–7, 2025.
6. M. R. R. Deva and N. Jain, “Utilizing Azure Automated Machine Learning and XGBoost for Predicting Cloud Resource Utilization in Enterprise Environments,” in 2025 International Conference on Networks and Cryptology (NETCRYPT), IEEE, May 2025, pp. 535–540. doi: 10.1109/NETCRYPT65877.2025.11102235.
7. Q. T. Sadat, G. Singh, N. Adik, R. S. Bhandari, and V. Kumar, “AI-Empowered IoT Frameworks for Predictive Maintenance and Knowledge-Driven Process Optimization,” in 2025 International Conference on Computational Engineering, Sensing Technology and Management (ICCETM), Sydney, Australia: IEEE, Nov. 2025, pp. 1–6. doi: 10.1109/ICCETM66557.2025.11557868.
8. V. Prajapati, “Advances in Software Development Life Cycle Models: Trends and Innovations for Modern Applications,” J. Glob. Res. Electron. Commun., vol. 1, no. 4, pp. 1–6, April, 2025, doi: 10.5281/zenodo.15222129.
9. A. R. Duggasani, “The Integration of Java-Based Applications in Software Development and AI: A Conceptual Review,” Int. J. Adv. Res. Sci. Commun. Technol., vol. 5, no. 1, pp. 460–471, Sep. 2025, doi: 10.48175/IJARSCT-28959.
10. G. Maddali, “Efficient Machine Learning Approach Based Bug Prediction for Enhancing Reliability of Software and Estimation,” SSRN Electron. J., vol. 8, no. 6, 2025, doi: 10.2139/ssrn.5367652.
11. M. Venkatram, “Leveraging Ai Models For Proactive Problem Detection, Investigation, And Root Cause Analysis In Enterprise It Infrastructure,” Int. J. Inf. Technol., vol. 6, no. 1, pp. 43–57, Jun. 2025, doi: 10.34218/IJIT_06_01_005.
12. A. Joon, “Smart Predictive Maintenance: AI-Driven Adaptation for Industrial Equipment,” in 2025 IEEE North-East India International Energy Conversion Conference and Exhibition (NE-IECCE), IEEE, Jul. 2025, pp. 1–6. doi: 10.1109/NE-IECCE64154.2025.11183108.
13. G. Modalavalasa, “Exploring Big Data Role in Modern Business Strategies: A Survey with Techniques and Tools,” Int. J. Adv. Res. Sci. Commun. Technol., vol. 3, no. 3, pp. 431–441, Jan. 2023, doi: 10.48175/IJARSCT-11900B.
14. M. De la Cruz Cabello, T. P. Sales, and M. R. Machado, “Log anomaly detection in AIOps: A real-world implementation using Large Language Models,” Syst. Soft Comput., vol. 8, p. 200475, Jun. 2026, doi: 10.1016/j.sasc.2026.200475.
15. R. Q. Majumder, “Machine Learning for Predictive Analytics: Trends and Future Directions,” Int. J. Innov. Sci. Res. Technol., vol. 10, no. 4, pp. 3557–3564, 2025, doi: 10.38124/ijisrt/25apr1899.
16. S. Garg, “Next-Gen Smart City Operations with AIOps & IoT : A Comprehensive look at Optimizing Urban Infrastructure,” J. Adv. Dev. Res., vol. 12, no. 1, 2021, doi: 10.5281/zenodo.15364012.
17. N. Prajapati, “The Role of Machine Learning in Big Data Analytics: Tools, Techniques, and Applications,”

ESP J. Eng. Technol. Adv., vol. 5, no. 2, pp. 16–22, 2025, doi: 10.56472/25832646/JETA-V5I2P103.

18. S. K. Tiwari and F. Investments, "Integration of AI and Machine Learning with Automation Testing in Digital," Int. J. Appl. Eng. Technol., vol. 5, no. 1, 2025.

19. K. Zhang, J. Xu, M. R. Min, G. Jiang, K. Pelechrinis, and H. Zhang, "Automated IT system failure prediction: A deep learning approach," in 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA: IEEE, Dec. 2016, pp. 1291–1300. doi: 10.1109/BigData.2016.7840733.

20. N. Y. Mulongo, "Key Performance Indicators of Artificial Intelligence For IT Operations (AIOPS)," in 2024 International Symposium on Networks, Computers and Communications (ISNCC), IEEE, Oct. 2024, pp. 1–8. doi: 10.1109/ISNCC62547.2024.10759030.

21. S. Prasetio, N. Raharya, A. S. Prabuwono, and G. Wibisono, "GenPdM: A Predictive-to-Decision Framework Integrating LSTM-Based Time-to-Next-Incident Forecasting and Retrieval-Augmented Generation for IT Service Operations," in IEEE Access, IEEE, 2026, pp. 130947–130964, August. doi: 10.1109/ACCESS.2026.3726289.

22. M. Joy et al., "AIOps in Action: Streamlining IT Operations Through Artificial Intelligence," Mar. 2025, International Journal of Intelligent Systems and Applications in Engineering. doi: 10.2139/ssrn.5257975.

23. M. Singh, "Enhancing Site Reliability Engineering Through AIOps: A Framework for Next-Generation IT Operations," Asian J. Res. Comput. Sci., vol. 18, no. 4, pp. 272–284, Mar. 2025, doi: 10.9734/ajrcos/2025/v18i4619.

24. B. K. Salman Khan, "Artificial Intelligence in IT Operations: AIOps for Predictive Analytics and Automation," Int. J. Mach. Learn. Res. Cybersecurity Artif. Intell., vol. 15, no. 1, 2024.

25. Y. Jani, "AI and Machine Learning for Predictive Monitoring in IT Operations," Int. J. Sci. Res., vol. 12, no. 12, pp. 2559–2140, Dec. 2023, doi: 10.21275/SR24822171836.

26. Q. Cheng et al., "AI for IT Operations (AIOps) on Cloud Platforms: Reviews, Opportunities and Challenges," 2023. doi: 10.48550/arXiv.2304.04661.

27. S. C. Pallaprolu and K. M. Hussain, "Predictive Modeling and Machine Learning-Based Logistics Fleet Operations Optimization," in 2025 IEEE Madhya Pradesh Section Conference (MPCON), IEEE, Aug. 2025, pp. 309–315. doi: 10.1109/MPCON66082.2025.11256508.

28. S. V. Khond, "Effect of Data Normalization on Accuracy and Error of Fault Classification for an Electrical Distribution System," Smart Sci., vol. 8, no. 3, pp. 117–124, Jul. 2020, doi: 10.1080/23080477.2020.1799135.

29. D. A. Bhanage, A. V. Pawar, and K. Kotecha, "IT Infrastructure Anomaly Detection and Failure Handling: A Systematic Literature Review Focusing on Datasets, Log Preprocessing, Machine & Deep Learning Approaches and Automated Tool," IEEE Access, vol. 9, pp. 156392–156421, 2021, doi: 10.1109/ACCESS.2021.3128283.

30. L. N. Valli, N. Sujatha, and E. J. Rathinam, "A Study on Deep Learning Frameworks to Understand the Real Time Fault Detection and Diagnosis in IT Operations with AIOPs," in 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT), Bengaluru, India: IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/EASCT59475.2023.10392382.

31. G. P. Bhandari, R. Gupta, and S. K. Upadhyay, "An approach for fault prediction in SOA-based systems using machine learning techniques," Data Technol. Appl., vol. 53, no. 4, pp. 397–421, 2019, doi: 10.1108/DTA-03-2019-0040.

32. S. Kumar, "Proactive System Maintenance: Machine Learning Models for Predicting and Preventing Potential System Failures," Int. J. Res. Appl. Sci. Eng. Technol., vol. 12, no. 5, pp. 1846–1851, May 2024, doi: 10.22214/ijraset.2024.61922.

33. S. K. Jangam and N. Karri, "Potential of AI and ML to Enhance Error Detection, Prediction, and Automated Remediation in Batch Processing," Int. J. AI, BigData, Comput. Manag. Stud., vol. 3, no. 4, pp.

70–81, 2022, doi: 10.63282/3050-9416.IJAIBDCMS-V3I4P108.

34. A. Wahid, J. G. Breslin, and M. A. Intizar, “Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework,” Appl. Sci., vol. 12, no. 9, p. 4221, Apr. 2022, doi: 10.3390/app12094221.