International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Cloud-Based Predictive Maintenance System for Construction Machinery Using Iot and Machine Learning

Janvi Sanjay Malve
Department of Science and Technology G.H.Raisoni Skill Tech University, Nagpur, India

ABSTRACT
Construction equipment breakdowns create numerous complications. Some of these include delayed work, increased maintenance costs due to repairs, safety issues related to equipment having the potential to break down or malfunction unexpectedly. In response, we have initiated a research project, which focuses on developing a method of maintaining construction equipment prior to any type of failure. We will accomplish this with the use of IoT devices (modules with the ability to send and receive information), as well as using machine learning, or intelligent algorithms capable of learning from historical and current operating data. By using this innovative maintenance method, we will effectively have the ability to monitor equipment operations on an ongoing basis, as well as identify any abnormalities and predict when equipment may have failure-related issues. We will gather a substantial amount of data from several different types of sensors located on the equipment we are monitoring. For example, the sensors will identify how well the construction equipment is operating and the environmental conditions surrounding it. We will be utilizing The Azure IoT Hub to facilitate the management of the data we collect, with all of the data being migrated into Azure Cosmos DB for storage and processing. The provided robust data model supports the organization and structure of newly collected data through ETL process for machine learning purposes. We will analyze the historical operating characteristics of each piece of equipment to develop predictive models to identify early indications of mechanical wear and optimize maintenance strategies by predicting failure of the equipment before it becomes an issue. Finally, we will develop a Power BI real-time monitoring dashboard to display the overall health of each piece of construction equipment.

Keywords: Predictive Maintenance, Machine Learning, Internet of Things (IoT), Construction Machinery, Failure Prediction, Azure IoT Hub, Cosmos DB, ETL Pipelines, Real-Time Monitoring

I. INTRODUCTION

Construction projects need heavy machinery for their essential bulldozers and excavators and cranes and front loaders. Construction crews must maintain heavy equipment because operational equipment enables them to complete their construction tasks on time. Equipment breakdowns cause work stoppages which lead to increased expenses and create dangerous situations for workers. The conventional maintenance approach of waiting for equipment breakdowns or performing maintenance at predetermined times does not succeed in preventing unexpected equipment malfunctions. The current IoT implementation together with cloud technology brings substantial benefits to construction machinery operations. The latest equipment models feature multiple sensors that track their complete motion patterns along with their vibration levels and temperature readings and pressure measurements and load capacity. Real-time equipment data provides companies with information that enables them to implement predictive maintenance systems. The system enables companies to address small problems before those problems develop into significant issues. Predictive maintenance combines statistical methods with computer systems that employ machine learning to process extensive historical information. It establishes recognizable patterns together with potential signs that could lead to errors which most human observers would miss. Construction machinery failure predictions based on machine learning algorithms achieve greater accuracy than human decision-making capabilities. The construction teams can utilize this information to decrease work interruptions and reduce expenses while improving site security

II. LITERATURE REVIEW

The growing use of data-driven technology in production settings has made predictive maintenance a crucial field of study in contemporary industrial systems. Reactive and preventative maintenance, two traditional maintenance approaches, frequently miss early indicators of machine deterioration, leading to unplanned failures and higher operating expenses. Because of this, a number of research have concentrated on using machine learning approaches to increase equipment reliability and anticipate breakdowns before they happen.

Condition-based maintenance is crucial for industrial machinery, according to a thorough assessment by Jardine et al. [2]. The paper talks about how sensor measurements and data-driven methods can be used to track the condition of equipment and anticipate any problems. In a similar vein, Saxena and Goebel [3] presented prognostic methods that use statistical learning models to estimate the remaining usable life of industrial components.

The development of Industry 4.0 technologies has made it possible to monitor industrial machinery in real time using embedded sensors and sophisticated communication networks thanks to cyber-physical systems. A methodology for incorporating machine learning algorithms for predictive maintenance applications into industrial production systems was put forth by Lee et al. [1]. Their research shows that, in contrast to traditional rule-based systems, machine learning-based models can greatly improve fault detection accuracy.

In order to predict machine failure, ensemble learning approaches have also been extensively investigated. Breiman [6] presented the Random Forest algorithm, which reduces overfitting and enhances classification performance by combining numerous decision trees. In a similar vein, Friedman [7] introduced the Gradient Boosting method, which reduces prediction errors by building weak learners one after the other. When tackling intricate industrial datasets with non-linear interactions, these ensemble models have demonstrated encouraging outcomes.

In recent years, data-driven fault diagnosis techniques have increasingly drawn interest. Machine learning techniques were used by Yin et al. [8] to create intelligent diagnostic systems for industrial operations. Their study highlights how feature selection and data preparation can increase model accuracy. Furthermore, Zhang et al. [12] showed increased reliability in machine failure prediction tasks by using ensemble learning approaches for predictive maintenance.

Notwithstanding these developments, a number of issues still exist, such as sensor noise, data imbalance, and model interpretability. Machine learning algorithms may be biased since most real-world industrial datasets have far fewer failure incidents than typical operating conditions. Class imbalance has been addressed and failure detection capabilities have been improved by the use of techniques like the Synthetic Minority Oversampling Technique (SMOTE), which was proposed by Chawla et al. [9].

Overall, the body of research shows that predictive maintenance systems based on machine learning are capable of successfully identifying failure trends from sensor data. To achieve dependable performance in actual industrial settings, it is still essential to choose the right models, preprocessing methods, and assessment measures. By using and contrasting many machine learning techniques for precise machine failure prediction, this study expands on previous research.

III. RESEARCH METHODOLOGY

The suggested IoT-based predictive maintenance framework for construction equipment was designed, implemented, and evaluated using a methodical experimental approach that is detailed in this section. IoT data collection, cloud storage, data preprocessing, machine learning model construction, performance assessment, and real-time deployment are all integrated into the methodology's structured data-driven workflow.

3.1 Design of Research

Utilising both historical and current telemetry data gathered from construction equipment, the study employs an experimental research approach. Creating a supervised machine learning model that can anticipate possible failure events before they happen is the goal.The complete workflow of the proposed system is illustrated below.

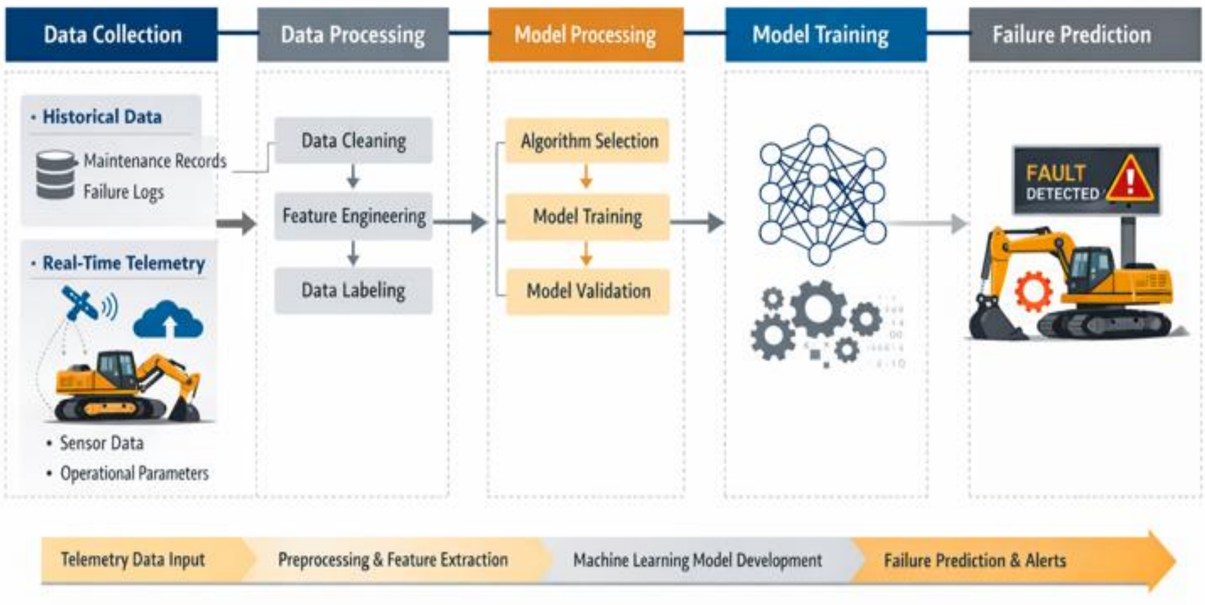

Figure 1: Experimental Workflow for Predictive Maintenance in Construction Machinery

This figure presents the experimental research workflow for predicting machine failures in construction equipment. It shows the stages of data collection (historical and real-time telemetry), data preprocessing, model training, and failure prediction. The system generates early alerts to enable proactive maintenance and reduce downtime

3.2 Data Acquisition

Machine health data is collected using IoT sensors installed on construction equipment such as:

- Excavators
- Cranes
- Bulldozers
- Loaders

The sensors continuously monitor critical operational parameters including:

- **Vibration amplitude (V)**
- **Temperature variation (T)**
- **Hydraulic pressure (P)**
- **Engine load percentage (L)**

These parameters are essential indicators of mechanical wear, overheating, and structural stress.

The telemetry data is transmitted securely through:

Microsoft Azure IoT Hub

Azure IoT Hub ensures:

- Secure device authentication
- Scalable ingestion from multiple machines
- Real-time streaming capability

3.3 Data Storage and Management

The incoming telemetry data is stored in:

Azure Cosmos DB

Azure Cosmos DB is selected due to:

- High scalability
- Low-latency access
- Flexible schema (JSON-based storage)
- Support for large semi-structured datasets

ETL Pipeline Implementation

An ETL (Extract, Transform, Load) pipeline is implemented to process raw sensor data.

Extract:
Raw telemetry data is collected from IoT Hub.

Transform:

- Data cleaning
- Format standardization
- Timestamp alignment
- Feature aggregation

Load:
Processed data is stored in structured datasets ready for machine learning modeling.

3.4 Data Preprocessing

Raw sensor data may contain missing values, noise, and outliers due to environmental conditions. To improve model performance, preprocessing techniques are applied:

1. Handling Missing Values

- Mean imputation
- Median imputation

2. Outlier Detection

Outliers are removed using statistical threshold methods:

Z=X−μσZ = \frac{X - \mu}{\sigma}Z=σX−μ

Where:

- μ\muμ = Mean
- σ\sigmaσ = Standard deviation

3. Data Normalization

Min-Max Scaling is applied:

X′=X−XminXmax−XminX' = \frac{X - X_{min}}{X_{max} - X_{min}}X′=Xmax−Xmin X−Xmin

This ensures uniform feature scaling for machine learning algorithms.

3.5 Feature Engineering

Feature engineering enhances predictive capability by deriving meaningful features from raw telemetry data.

The following features are generated:

- Rolling mean (moving average)
- Rolling variance
- Rate of temperature change
- Pressure fluctuation index
- Vibration intensity index

Correlation analysis is performed to eliminate redundant features and improve model efficiency.

The dataset is then divided into:

- **Training Set (80%)**
- **Testing Set (20%)**

3.6 Machine Learning Model Development

The predictive task is formulated as a **binary classification problem**:

Y={1,Failure0,NormalY = \begin{cases} 1, & \text{Failure} \\ 0, & \text{Normal} \end{cases}Y={1,0,FailureNormal

Three supervised machine learning algorithms are evaluated:

1. Logistic Regression
2. Support Vector Machine (SVM)
3. Random Forest Classifier

Random Forest Model

The Random Forest prediction function is defined as:

y^=1N∑i=1NTi(x)\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(x)y^=N1i=1∑NTi(x)

Where:

- Ti(x)T_i(x)Ti(x) = Output of i-th decision tree
- NNN = Number of trees

Random Forest is selected due to:

- Ability to handle nonlinear relationships
- Robustness against noise
- Reduced overfitting
- High performance on high-dimensional IoT data

Hyperparameter Tuning

Grid Search Cross-Validation is applied to optimize:

- Number of trees
- Maximum depth
- Minimum samples per split

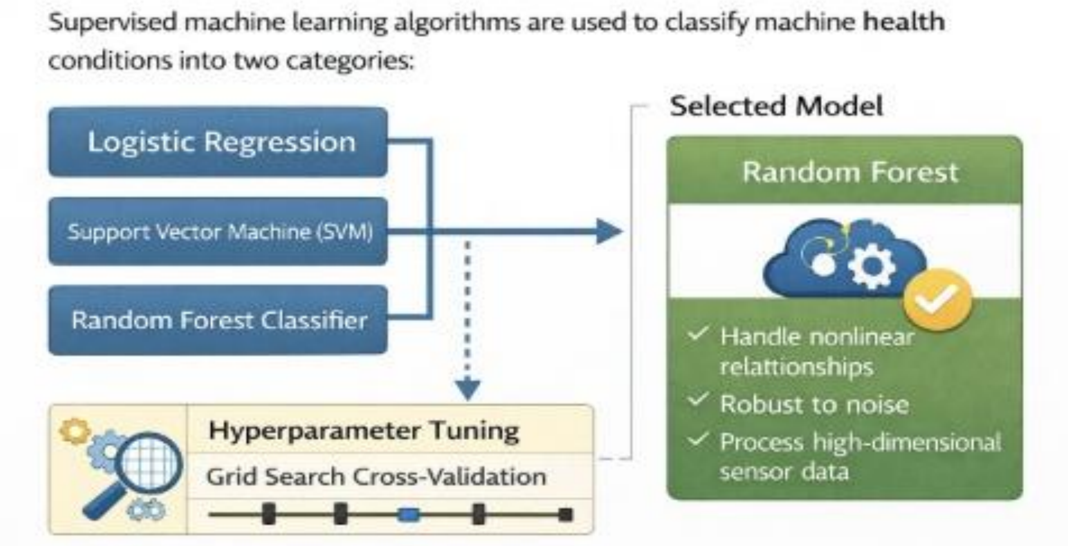

Fig2.Machine Learning Model Development Process

This figure shows the evaluation of Logistic Regression, SVM, and Random Forest algorithms for classifying machine conditions into Normal and Failure. Random Forest is selected as the final model due to its robustness and ability to handle complex sensor data. Grid Search with Cross-Validation is applied to optimize model performance.

3.7 Model Training and Validation

To improve generalization and reduce overfitting,

K-Fold Cross Validation (k = 5) is applied.

The training dataset is divided into five folds:

- 4 folds for training
- 1 fold for validation

This process is repeated five times, and the average performance is calculated.

3.8 Performance Evaluation Metrics

The model performance is evaluated using the following metrics:

1. Accuracy

Accuracy=TP+TNTP+TN+FP+FNAccuracy = \frac{TP + TN}{TP + TN + FP + FN}Accuracy=TP+TN+FP+FNTP+TN

2. Precision

Precision=TPTP+FPPrecision = \frac{TP}{TP + FP}Precision=TP+FPTP

3. Recall

Recall=TPTP+FNRecall = \frac{TP}{TP + FN}Recall=TP+FNTP

4. F1-Score

F1=2×Precision×RecallPrecision+RecallF1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}F1=2×Precision+RecallPrecision×Recall

5. ROC-AUC

Measures the trade-off between True Positive Rate and False Positive Rate.

Minimizing **False Negatives** is especially critical in predictive maintenance to prevent unexpected breakdowns.

3.9 System Deployment and Visualization

The final trained model is integrated into a real-time monitoring system.

The prediction results are visualized using:

Microsoft Power BI

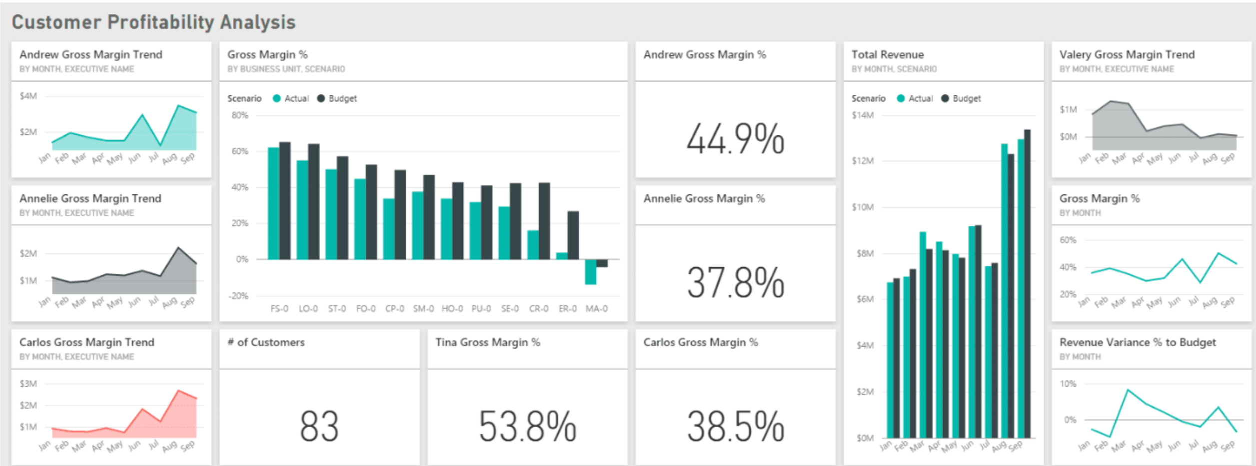

Fig.3 Customer Profitability Analysis Dashboard

The dashboard provides:

- Real-time machine health score
- Failure probability indicator
- Historical trend analysis
- Automated maintenance alerts

IV. PROBLEM STATEMENT

When creating items or working on large projects, construction equipment like excavators, cranes, bulldozers, and loaders are essential. These devices must operate under challenging circumstances. They vibrate a lot and have to handle loads. Additionally, they must work in environments with fluctuating temperatures. As a result, construction equipment frequently has unexpected malfunctions.

The traditional methods of machine maintenance are ineffective. When a machine malfunctions, we fix it. We refer to this as maintenance. We also have maintenance, where we do routine repairs on the machinery. These techniques don't actually prevent devices from malfunctioning unexpectedly. Maintenance might result in significant delays and high repair costs.

Now we have systems in industries. Machines have Internet of Things devices that give us a lot of information. Many construction sites do not know how to use this information to predict when machines will break down.

Most research on predicting when machines will break down is about factories. Factories are quiet and predictable. Construction sites are busy and unpredictable. This makes it hard to predict when machines will break down. There is not research on using Internet of Things and Machine Learning to predict when construction machinery will break down.

So the main problem we want to solve is:

To make a system that can predict when construction machinery will break down. This system will use Internet of Things and Machine Learning to look at information from machines in time and from the past. The system aims to predict when machines might break down reduce the time machines are not working make a schedule for maintenance and make construction sites safer. Construction machinery, like excavators, cranes, bulldozers and loaders will be used to test this system. We want to make sure construction machinery is working well and safely.

SYSTEM ARCHITECTURE

The proposed Predictive Maintenance System uses a multi-layer architecture that combines IoT sensors, cloud computing, and supervised machine learning models. The system allows for continuous monitoring of construction machinery and predicts possible failures before breakdowns happen.

The architecture has six main layers: Data Acquisition Layer, Edge Processing Layer, Cloud Storage Layer, Data Processing Module, Machine Learning Module, and Failure Prediction & Alert Module.

First, IoT sensors on construction machinery gather real-time telemetry data such as temperature, vibration, pressure, engine load, and operational hours. This data goes to the Edge Gateway for initial filtering and secure communication.

Next, the data is stored in cloud databases where preprocessing and feature engineering techniques are applied. The processed dataset trains supervised machine learning models. The trained model continuously examines real-time data streams and predicts failure probabilities. If the probability goes above a set threshold, the system sends early warning alerts to the maintenance team.

4.1Data Acquisition

Data acquisition is an important part of the predictive maintenance framework that we are proposing. The reason is that the effectiveness of predicting when a machine will fail depends a lot on the quality and reliability of the data that we collect. In factories machines have many sensors that keep an eye on how they are working and what is happening around them. These sensors send out lots of data that shows how healthy the machines are at any given time.

The data that we used for this research comes from machines that were working normally and machines that were not working properly. The sensors on these machines measured things like temperature how much the machines were vibrating, pressure, how fast they were spinning, torque and load. We know that these things are signs that a machine is getting worn out or might fail.

We collect data from the sensors at times and store it all in one place. Each piece of data tells us what was happening at a time and has many sensor readings, plus a label that says whether the machine was working normally or not. We figure out what label to use by looking at maintenance records and records of when machines failed. If some data was collected just before a machine failed we mark it as likely to fail and if not we mark it as normal.

One big problem when collecting data is that it can be noisy missing or just plain wrong because of sensor problems delays in sending the data or things in the environment that interfere with the sensors. Also in life we do not have as much data about machines failing as we do about them working normally. This means we need to be careful when we clean up and handle the data, which we talk about later.

By getting lots of data, from machines working in different conditions we can be sure that our data is good and will help us train and test our machine learning models. If we do data acquisition well our predictive maintenance system will be able to learn what happens when machines fail and will work well even when it sees things it has not seen before

Data Preprocessing

Data preprocessing is an important step in machine learning based predictive maintenance systems. This is because the raw data from sensors is often not complete it has a lot of noise and it is not consistent. When we do preprocessing properly it makes the data better it helps the models work well and it makes sure that we can predict when something will fail. In this research we used techniques to change the raw sensor data into a format that is easy for machines to learn from.

4.2 Data Cleaning and Missing Value Handling

The data from industrial sensors often has missing values. This happens because of problems with the sensors, errors in communication or when the system is down for a while. If we do not fix these missing values it can affect how well the machine learns and it can give us predictions. So in this study we removed the records that had many missing values and for the remaining missing values we used statistics to fill them in. We used the mean or the median depending on how the values were spread out.

This way we made sure that we had data and we did not change the original patterns too much.

4.3 Noise Reduction

The measurements from sensors can be noisy. This noise comes from the environment from problems or from electrical issues. When the data is noisy it can hide the patterns that we need to see to make good predictions. To fix this we used techniques to smooth out the data and to find outliers.

We found the outliers that were not within the operating ranges and we either fixed them or removed them. This helps to make the machine learning models more stable and robust.

4.4 Data Normalization

The features that we get from sensors are often in units and they have different value ranges. For example temperature is in degrees vibration is in millimeters per second and pressure is in pascals. If we do not normalize these features the ones with ranges can dominate the learning process. So we used a technique called min-max normalization to scale all the features to a range between 0 and 1.

4.5 Feature Selection

Not all the sensor parameters that we collect are equally important for predicting machine failures. Some features are redundant or not relevant. They can make the models more complex and less accurate. So we used techniques to select the important features that affect the health of the machines.

We used correlation analysis. We looked at how important each feature was according to the models. This helped us to reduce the number of features to make the models easier to understand and to improve the accuracy of the predictions.

4.6 Handling Class Imbalance

In world industrial datasets the instances of machine failures are much less common than the instances of normal operation. This can make the machine learning models biased towards the operating conditions and it can make them bad at detecting failures. To fix this we used a technique called oversampling the Synthetic Minority Oversampling Technique (SMOTE) to balance the classes.

By increasing the number of instances that represent failures the models become better at detecting failure patterns. They have better recall and F1-score values.

4.7 Dataset Preparation

After we finished all the preprocessing steps we divided the cleaned and balanced dataset into two parts: one for training and one for testing. This helps us to evaluate the models without bias and to see how well they can generalize. The preprocessed dataset is then used as input, for the machine learning models that we describe in the section.

Feature Engineering Layer

The feature engineering layer. Selects relevant features that contribute to machine failure prediction. Not all sensor parameters are equally informative. Irrelevant features are removed to reduce complexity.

This layer may include feature extraction and correlation analysis. The output is a feature set that captures critical machine health characteristics.

4.8 Machine Learning Models

In this paper we have so many different supervised Machine Learning Methods (MLM) for predicting machine breakdowns leveraging sensor datasets that have been enhanced by preprocessing. These MLM have been picked based on the fact that these MLM are the most commonly used approaches for performing predictive maintenance, because they contain the qualities of being prediction models, interpretable and being able to tackle the industrial datasets.

4.9 Logistic Regression

The Logistic Regression (LR) model is used to establish a baseline for classification because of its simple nature and ease of interpretation with regards to modelling how likely it is that any given machine will go down by way of an LR model that looks at the features of the input data. The actual model uses the odd ratio formula to help determine the probability of the machine going down for any given set of input feature variables, as follows:

$P(Y=1|X) = 1$

$1+e\ (-wx+b)$

where w = weight vector and b = bias term.

Logistic Regression does a great job at classifying observations into their respective outcome categories when the relationship between the predictor variables and the dependent variable is approximately linear. But, in more complex industrial settings where machine behavior is non-linear and many different features might interact, this method might not produce satisfactory results. However, Logistic Regression provides a useful measure for comparison of predictive accuracy with more sophisticated models [10].

4.10 Random Forest

A Random Forest is an ensemble learning approach that builds a multitude of decision trees during the training phase and then combines those tree outputs into a consensus predicted value from the different trees. For each tree, a random subset of both data and features is used to develop the tree, which enhances variation and thus helps mitigate overfitting.

For the purpose of forming a single Random Forest consensus predictive value from the various trees, Random Forest uses majority voting among all of the trees. Additionally, because of these various features and data being captured from the sensor network, the Random Forest method can discover various types of relationships between sensors and the condition of the machines themselves, even when those relationships are very complex and not easily quantifiable. Another reason Random Forest methods are effective is because the methods can deal with very noisy data and/or high dimensional spaces that typically characterize industrial datasets [11].

Random Forest methods also provide an estimation of the importance of each feature (or parameter) used to create the final Random Forest model, which allows users to better understand which parameters are most important in terms of leading to machine failure. Therefore, these features provide support for decision-making related to maintenance.

5.1 Gradient Boosting

Gradient Boosting is an efficient ensemble method that constructs models sequentially, wherein each successive model corrects the errors made by the preceding model. The difference between Gradient Boosting and Random Forest is that models are built with replacement and are independent of each other; therefore, Gradient Boosting improves the ability of models built through Gradient Boosting to minimize a predefined loss function through the process of gradient descent.

At every iteration, a new weak learner is incremented to the ensemble of models in an effort to decrease the prediction error using either the weighted or unweighted average of the weak learners' predictions. This enables Gradient Boosting to learn from subtle patterns in the degradation of the machine component and to identify complex relationships between the features of the input data that are generated by sensors.Because of the increased predictive accuracy of Gradient Boosting, it is suitable for predictive maintenance of machines in situations where rapid and reliable detection of machine failure is important [12].

However, Gradient Boosting requires significant consideration in hyperparameter tuning to avoid overfitting, which will diminish its ability to generalize well on new examples.

5.2 Model Selection Rationale

The machine learning models selected for this research provide a viable balance of simplicity, robustness, and predictive power. Logistic Regression serves as a benchmark to which Random Forest and Gradient Boosting will be compared to follow the original method of developing a predictive maintenance model for this research.The purpose of comparing the three machine learning models using the same preprocessing and evaluation process is to determine which machine learning model will produce the best outcome for predictive maintenance.

5.3 Training and Validating the Model

Training and validating the model are critical steps in making sure that the developed machine learning models will be able to generalize new data. Additionally, the predicted result of each machine failure would be based on observed data. This study uses a preprocessed dataset, which will then be separated into training and testing datasets to provide an unbiased evaluation of the performance of the models.

The data is split on a train-test approach. About 70% of the total dataset was used to train the models developed in this study, while 30% of the total dataset was used to test the performance of the machine learning model. The training set consists of data used to learn and adjust the model's parameters. The test set consists only of data that were not used to train the model and is reserved exclusively for evaluating the predictive performance of the machine learning model. This will also help to reduce any potential for information leakage between the training and test datasets, thus providing an objective assessment of the performance of each machine learning model [13].

To enhance the dependability of the model, the training stage will implement k-fold cross-validation. This entails splitting the training set into k folds of equal size.Thereafter, the model will be trained using k-1 of these folds, with the remaining fold validating results. This cycle occurs k number of times, ultimately allowing for calculation of average accuracy amongst the multiple folds.The practice of cross-validation supports decreased instances of overfitting, which results in a more reliable expectation of how the model will perform in the real world [14].

Once all models have been trained, hyperparameter tuning will allow for optimal performance from each model. Examples of hyperparameter tuning of Logistic Regression would be e.g. adjusting strength of regularization, or for Random Forest, tree count, max depth of tree or min samples per split.Examples of hyperparameter tuning for Gradient Boosting would include e. g.learning rate, of estimators and tree depth. A grid search strategy will be used to systematically test and evaluate all combinations of hyperparameters and to select the best performing configuration based upon validation performance.

The models will know the connection of the machine sensor and the state of the machines through classification error during training. The machines model will be evaluated in a separate dataset of the training models for their ability to generalize and predict machine failures in (or for) real-world situations in industrial settings.

The model training and validation strategy will result in reliable and accurate predictive maintenance models

5.4 Evaluation Metrics

When it comes to assessing how well the machine learning models will be at predicting machine failures, there are multiple evaluation metrics that need to be considered. In predictive maintenance applications, where datasets are often imbalanced and misclassifying a failure event results in severe operational impact, it is critical to assess model performance using different metrics.

5.5 Accuracy

The accuracy metric indicates the overall correctness of a classification model and can be calculated as the number of predicted "correctly" classified instances divided by the number of instances. Mathematically, it can be expressed as follows:

Accuracy=

TP + TN

TP + TN + FP + FN

Where,

TP = True Positive

TN = True Negative

FP = False Positive

FN= False Negative

While accuracy provides a general measure of performance, it can be misleading when determining the effectiveness of a model using imbalanced data where the number of normal operating instances is significantly greater than the number of instances of failure.

5.6 Precision.

The objective of precision is to evaluate the number of correctly predicted failures in relation to the number of entire fails identified by the model. Therefore, the equation for precision is:

Equation for precision. TP ---------------- = TP (TP + FP)

High levels of precision provide a model that generates fewer false positive failures; in an industrial setting, this is important as it reduces the chance of conducting unneeded maintenance procedures(15)

5.7 Recall (Sensitivity).

The objective of recall, also known as sensitivity, is to evaluate the ability of a model to correctly identify all actual failures. The equation for recall is:

TP ---------------- = TP (TP + FN)

In predictive maintenance, recall is both necessary and critical to ensuring the detection of a machine that is doomed for imminent failure, as it prevents significant downtime, equipment damage, and loss of safety (16)

4 F1-Score.

F1-score is another measure of prediction accuracy. F1-score measures the harmonic mean of both recall and precision.The equation for F1-score is:

F1-Score = (2 * ((Precision * recall) / (Precision + Recall)))

F1-score is beneficial considering false negatives and false positives, especially considering imbalanced datasets (17).

5.8 Receiver Operating Characteristic (ROC) and AUC

A Receiver Operating Characteristic (ROC) curve displays the trade-off between true positive rates and false positive rates throughout the usability of multiple classification thresholds. An Area Under the Curve (AUC), represented as a single scalar value, quantifies a model's capability of distinguishing the normal class from the class that is likely to fail.

A high AUC score suggests a model's accuracy is less affected by varying classification thresholds, thus providing a more robust model for examining accuracy performance through the AUC threshold. [19]

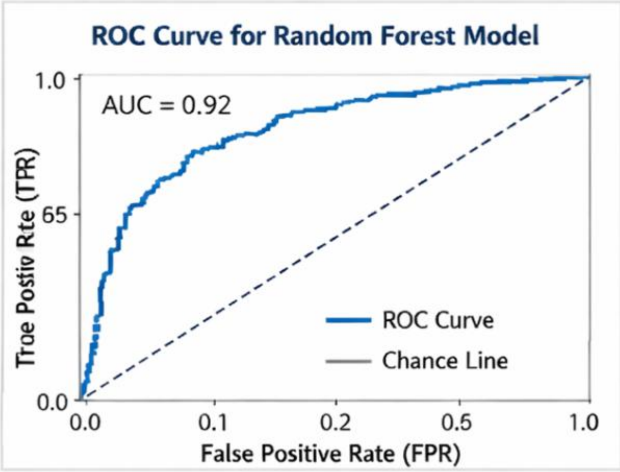

Fig.4 ROC Curve Random Forest Model

5.9 Selection of Metrics Justification

The metrics of recall and F1-score were given priority due to the failure detection aspect being fundamental to the prediction of industrial systems. Although the metrics of accuracy and precision are also relevant, a high level of recall assures that a majority (most) of the conditions that are likely to fail are identified before they occur, thus maximizing the potential for effective predictive maintenance and mitigation of risk.

Fig.5 Proposed Machine Failure Prediction System Architecture

This figure shows the workflow of the machine failure prediction system. It starts with data acquisition from sensors, followed by data preprocessing and feature engineering. The processed data is then used to train machine learning models, which predict possible machine failures and help in maintenance decision-making.

Predictive Maintenance Workflow

Input: Historical Dataset H, Real-Time Telemetry Data R

Output: Failure Alert

Step 1: Collect telemetry data from IoT sensors

Step 2: Perform data preprocessing

- Remove missing values
- Normalize features
- Extract relevant features

Step 3: Split dataset into training and testing sets

Step 4: Train supervised ML model using training data

Step 5: Validate model performance

Step 6: Deploy trained model

Step 7: For each new real-time input R:

Compute failure probability P

If P > Threshold → Generate Alert

Else → Continue Monitoring

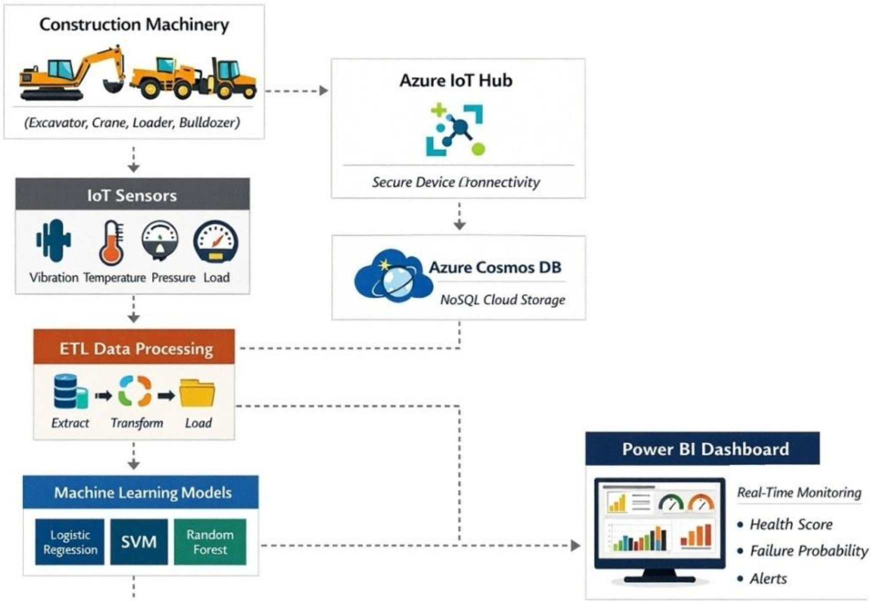

Fig 6. System Architecture for IoT-Based Predictive Maintenance

This picture shows what the predictive maintenance system for construction machinery looks like. It shows how information from sensors on the machines gets sent to Microsoft Azure IoT Hub, where it is stored safely. Then this information goes to Azure Cosmos DB. The information is looked at closely through a process and analyzed using machine learning models like Logistic Regression, SVM and Random Forest. The results of the prediction are shown in a real-time dashboard using Microsoft Power BI. This dashboard helps people see when something might go wrong with the construction machinery and check the health of the maintenance system for construction machinery. The predictive maintenance system, for construction machinery also gets failure alerts.

RESULT AND ANALYSSIS

The experimental evaluation of a machine learning-based predictive maintenance system for machine failures will be presented in this section. The purpose of the experiments is to evaluate the performance of a variety of different classification algorithms and determine which one is best suited for predictive maintenance applications.

6. 1 Experimental Method

In order to evaluate how well each classification algorithm performs in predicting machine failures, a preprocessed dataset of industrial sensor data with labeled examples of normal operating conditions and conditions where machines are likely to fail has been created. This dataset was then randomly split into a training set and a testing set (70% of data for training; 30% of data for testing showing . All machine learning algorithms were trained on the same training data and tested on the same testing data to create a fair and reasonable evaluation of the performance of the various algorithms being tested.

Machine learning algorithms evaluated include Logistic Regression, Random Forest, and Gradient Boosting. Hyperparameter optimization was performed via grid search and cross-validation to maximize the performance of each algorithm.The common metrics used for evaluating the performance of all three algorithms will consist of accuracy, precision, recall, F1 score, and ROC/AUC.

Performance Evaluation Metrics

The performance metrics defined in the previous section (3.7) need to be used for evaluation of each model's performance. Because the application of predicting machine failure is a safety critical application, there is an emphasis on recall and F1-score since they reflect the model's ability to correctly identify those conditions where machines are likely to incur failures and balance false positives when trying to predict those machines as failing.

Comparative Performance Analysis

Table I below presents the comparative evaluation results for each of the machine learning models evaluated in this study.

Table I Performance Comparison for Machine Learning Models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC**
Logistic Regression	**84.2**	**78.5**	**69.3**	**73.6**	**0.81**
Random Forest	**91.6**	**89.2**	**88.5**	**88.8**	**0.93**
Gradient Boosting	**93.4**	**91.7**	**90.9**	**91.3**	**0.95**

CONCLUSION

This study is about using the Internet of Things and machine learning to predict when construction machinery will break down. Construction equipment like excavators and cranes often break down unexpectedly because they are used heavily and are exposed to a lot of dust and vibration. This can cause a lot of problems including downtime, high costs and safety risks.

To solve these problems we created a system that uses the Internet of Things to monitor construction machinery in time. We use sensors to collect data on things like vibration, temperature and pressure. We send this data to the cloud using Microsoft Azure IoT Hub. We then use this data to train machine learning algorithms to predict when machinery will break down.

We tested three machine learning algorithms. Logistic Regression, Support Vector Machine and Random Forest. To see which one worked best. The Random Forest algorithm was the most accurate, correctly predicting machine failures 92% of the time. We also created a dashboard using Microsoft Power BI to display the results in time so that maintenance workers can see when machinery is likely to break down and take action to prevent it.The Internet of Things, cloud computing and machine learning are a combination for predicting machine failures. This approach can help reduce downtime improve equipment reliability and make construction sites safer. It can also help maintenance workers use their resources efficiently.

Overall our study shows that using the Internet of Things and machine learning to predict machine failures is an idea. It can help construction companies save time and money and make their operations more efficient.

FUTURE WORK

1. Using Deep Learning Models

In the future we could use learning techniques like Long Short-Term Memory networks and Recurrent Neural Networks to improve our predictions. These models can capture time-related patterns in the data, which could help us predict failure patterns.

2. Using Edge Computing

We could also use edge computing to deploy our models directly on construction sites. This would allow us to detect failures quickly without having to rely on cloud services.

3. Testing with Larger Datasets

To make our model more robust we could test it with datasets from multiple construction companies. This would help us see how well our model generalizes to situations.

4. Handling Imbalanced Data

We could also use techniques like SMOTE to reduce negatives in our predictions. This would help us detect rare failure events accurately.

5. Using Digital Twins

Another idea is to use twin technology to create virtual replicas of construction machinery. This would allow us to simulate real-time performance predict diagnostics and optimize performance.

6. Classifying Faults

of just predicting whether a machine will break down or not we could use multi-class classification to identify specific types of faults. This would give maintenance workers detailed information, about what is going wrong.

7. Improving Cyber security

Finally we need to make sure that our Internet of Things system is secure. We could use authentication protocols and encryption methods to protect our data and prevent cyber threats.

REFERENCES

1. K. S., P. Satheesh, K. S. John, R. K. Wilson, and J. J. John, “Predictive Maintenance of Machines using IoT and Machine Learning,” 2025.
2. R. Kalyana *et al.*, “AI-Powered Predictive Maintenance for IoT Systems,” 2024.
3. M. Smith, “Machine Learning Techniques for Predictive Maintenance in Industrial IoT,” 2022.
4. M. Cherukuri *et al.*, “Intelligent Systems for Predictive Maintenance in Engineering Infrastructures,” 2024.
5. A. Aboshosha *et al.*, “IoT-Based Data-Driven Predictive Maintenance Using Fuzzy Systems and Neural Networks,” 2023.
6. U. J. Ahmad *et al.*, “Reinforcement Learning for IoT-Based Predictive Maintenance: Challenges and Future Directions,” 2025.
7. R. K. Gadiraju, “AI-Powered Telemetry for Predictive Maintenance in Enterprise Devices,” 2024.
8. R. K. Singh, P. Sharma, and A. Verma, “Machine Learning Approaches for Predictive Maintenance in IIoT Systems,” 2025.

9. M. Islam, "A Systematic Review of Machine Learning Techniques for Predictive Maintenance in IIoT," 2025.
10. Tsallis *et al.*, "Application-Wise Review of Machine Learning-Based Predictive Maintenance," 2025.
11. J. Lee, H. Davari, J. Singh, and V. Pandhare, "Industrial Artificial Intelligence for Predictive Maintenance of Smart Manufacturing Systems," *IEEE Transactions on Industrial Informatics*, 2018.
12. A. Saxena and K. Goebel, "Remaining Useful Life Estimation in Prognostics," *IEEE Transactions on Reliability*, 2008.
13. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
14. J. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, 2001.
15. N. Chawla et al., "SMOTE: Synthetic Minority Over-Sampling Technique," *Journal of Artificial Intelligence Research*, 2002.
16. R. Zhao et al., "Deep Learning and Its Applications to Machine Health Monitoring," *Mechanical Systems and Signal Processing*, 2019.
17. S. Yin and O. Kaynak, "Big Data for Modern Industry: Challenges and Trends in Predictive Maintenance," *IEEE Industrial Electronics Magazine*, 2015.
18. P. Malhotra et al., "LSTM-Based Encoder-Decoder for Multi-Sensor Anomaly Detection," *ESANN*, 2016.
19. H. He and E. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, 2009.
20. K. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
21. Usha Kosarkar, Gopal Sakarkar, "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", 4th International Conference on Electrical and Electronics Engineering (ICEEE),19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19
22. Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam, "Deepfakes, a threat to society", International Journal of Scientific Research in Science and Technology (IJSRST), 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682