Data Driven Insights into Elevator Maintenance: Predicting Vibration Patterns with IOT Sensor Technology

Injety Jayanth Das[1], Maku Akhilesh[2], Patil Hemeshwar Reddy [3], Mrs. K. Hemalatha[4]
[1,2,3] UG Scholar, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
[4]Assistant Professor, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
injetyjay@gmail.com

Abstract:

Elevators are a critical component of modern urban infrastructure, ensuring the smooth functioning of buildings and transport systems. Traditional elevator maintenance systems often rely on periodic inspections or reactive repairs, which may not detect emerging issues early enough, potentially leading to costly downtimes, safety concerns, or prolonged disruptions. As the demand for efficiency and safety in building operations rises, the integration of Internet of Things (IoT) sensors and machine learning in predictive maintenance has emerged as a promising solution. This study explores how IoT-enabled vibration sensors, paired with data-driven analytics, can enhance the predictive capabilities of elevator maintenance systems. Elevator maintenance has historically relied on scheduled service checks and reactive repairs following system failures or user complaints. However, these traditional methods often fail to predict mechanical issues before they result in breakdowns. The advent of IoT technologies in the early 21st century has revolutionized many sectors, including elevator systems, by offering continuous monitoring capabilities. Vibration sensors, in particular, have proven to be valuable tools for detecting early signs of mechanical wear and tear, misalignment, or imbalance in elevator components such as motors, cables, and bearings. Despite the advantages of IoT-based monitoring, elevators still often experience unexpected failures due to the inability of traditional systems to detect subtle, early-stage mechanical problems. This research aims to address the gap by using vibration data collected from IoT sensors to predict maintenance needs and prevent failures before they occur. The problem lies in the complexity of vibration signals, which require advanced analysis to distinguish between normal operational vibrations and those indicative of potential faults.

Keywords: Predictive Maintenance, IoT Sensors, Vibration analysis Elevator Monitoring, Machine Learning.

1. INTRODUCTION

Elevators are a critical component of modern urban infrastructure, ensuring the smooth functioning of buildings and transport systems. Traditional elevator maintenance systems often rely on periodic inspections or reactive repairs, which may not detect emerging issues early enough, potentially leading to costly downtimes, safety concerns, or prolonged disruptions. As the demand for efficiency and safety in building operations rises, the integration of Internet of Things (IoT) sensors and machine learning in predictive maintenance has emerged as a promising solution. This study explores how IoT-enabled vibration sensors, paired with data-driven analytics, can enhance the predictive capabilities of elevator maintenance systems. Elevator maintenance has historically relied on scheduled service checks and reactive repairs following system failures or user complaints. However, these traditional methods often fail to predict mechanical issues before they result in breakdowns. The advent of IoT technologies in the early 21st century has revolutionized many sectors, including elevator systems, by offering continuous monitoring capabilities. Vibration sensors, in particular, have proven to be valuable tools for detecting early signs of mechanical wear and tear, misalignment, or imbalance in elevator components such as motors, cables, and bearings. Despite the advantages of IoT-based monitoring, elevators still often experience unexpected failures due to the inability of traditional systems to detect subtle, early-stage mechanical problems. This research aims to address the gap by using vibration data collected from IoT sensors to predict maintenance needs and prevent failures before they occur. The problem lies in the complexity of vibration signals, which require advanced analysis to distinguish between normal operational vibrations and those indicative of potential faults.

2. LITERATURE SURVEY

Technological advancements in information technologies are progressively transforming our lives [**1**]. These changes demand an accelerated pace of management decision making [2,3]. As stated by the authors in [4], in the current scenario, producing an innovative product (or providing a service) that meets user requirements typically involves the integration of resources and competencies from multiple companies. The main finding from article [5] highlights the necessity for research and development in several crucial areas, including digital equipment maintenance and end-to-end automation, in order to enhance industries' preparedness for future problems.

Digital technologies and the Internet of Things (IoT) offer data homogeneity, distribution, editability, and the ability to self-reflect and reprogram, as stated in [**6**]. These features enable the implementation of multiple inheritance in distributed software applications, where no single owner possesses the entire design hierarchy or dictates the platform's core. As a result, products become open to new uses after manufacturing, as they can be arbitrarily combined through standardized interfaces [**7**].

The concept of condition-based maintenance (CBM) of industrial equipment [8] allows the determination of maintenance requirements and offers numerous benefits. CBM improves equipment credibility and dependability, and reduces maintenance resource costs compared to a late-scheduled maintenance approach. Under the CBM approach, maintenance is carried out only when specific metrics indicate declining performance or faults. The main problem with CBM is the need to spend significant resources on implementing equipment condition-monitoring tools, which usually include such non-invasive methods as visual inspection, and measurement of power consumption, noise, temperature, and vibration. Paper [9] proposes an integrated framework, which takes a broad perspective on CBM implementation, and integrates technological, organizational, and user-related elements. This study contributes to the field of CBM with a comprehensive view of implementation challenges and solutions in real-world implementations, from the original equipment manufacturer's (OEM's) point of view. Of the solutions proposed in article [10] for current research, we chose to prioritize the Use a state-of-the-art IoT platform for development; Define modular project-level software decisions; Outline methods for collaboration between hardware and software specialists. The authors of paper [11] emphasize the difference between Condition-based Maintenance (CBM) and Predictive-based Maintenance (PM) as two effective and complementary maintenance methods: CBM monitors the current condition; PM uses the CBM results to generate a future prediction for a machine.

Digital IoT platforms and end-to-end automation allow us to overcome the substantial resource consumption of the CBM concept. Well-known publications [12,13] extensively explore the development of cost-effective hardware and software solutions for vibration diagnostics using microelectromechanical systems (MEMS). A platform-oriented approach creates new possibilities for equipment fault diagnosis and state forecasting.

Smart sensors play a crucial role in CBM systems. According to the IEEE 1451.0-2007 standard [14], sensors with functions beyond the minimum required for measurements are classified as intelligent. Along with the digital interface and self-testing capabilities, these sensors have redundant functionality that simplifies their integration into networked applications.

In various mechanical systems, vibration diagnostics are an essential method for assessing the condition of mechanical systems, holding significant importance across multiple fields of application. Vibration is a highly versatile parameter that considers almost all aspects of a unit's state, allowing operating modes to determine the technical condition of the equipment.

The accelerometer manufacturer establishes the output characteristics following extensive testing, typically encompassing the influence of various operating conditions, such as temperature changes and magnetic fields. Paper [15] proposes a method for estimating the thermal behavior of capacitive MEMS accelerometers and compensating for their drifts in order to reduce orientation and temperature effects. It is a necessary solution, but insufficient for solving the general problem of compensating for accelerometer errors during regular operation.

The accelerometer metric of displacement at 0 g holds significant importance as it sets the baseline for measuring actual acceleration. Mounting the system with an accelerometer introduces additional measurement errors, which can arise due to stresses in the printed circuit board and the application of various compounds during mounting. As recommended in [16], we will calibrate after system assembly to exclude these errors.

The ISO 16063 series standards [17] set the modern requirements for vibration sensors and their calibration methods. Usually, a MEMS accelerometer calibration involves averaging the measurement values using a calibration scheme, where the accelerometer system is positioned to have one axis, typically the Z axis, experiencing a 1 g gravitational field, while the other axes, X and Y, remain in a 0 g field. After installation at a specific location, additional calibration is conducted by comparing the measurement results with those of a reference accelerometer [18].

3. PROPOSED METHODOLOGY

This project focuses on developing a predictive maintenance system for machinery, with a specific emphasis on analyzing elevator components. The system leverages machine learning models trained on IoT sensor data (e.g., vibration levels) to predict mechanical performance metrics, such as revolutions of a component. By predicting these metrics, the project aims to identify potential issues before they lead to system failures, enhancing operational efficiency.

Key Objectives

1. **Early Fault Detection**:
    - Predict mechanical performance metrics (e.g., revolutions) using IoT sensor data.
    - Detect subtle signs of wear and tear before significant failures occur.
2. **Data-Driven Insights**:
    - Analyze sensor data (e.g., vibration levels) to understand patterns associated with normal operation versus faults.
    - Provide actionable insights for maintenance teams.
3. **Efficiency Improvement**:
    - Reduce unplanned downtime by transitioning from reactive to predictive maintenance.
    - Optimize maintenance schedules based on data-driven predictions.
4. **Model Deployment**:
    - Build, train, and evaluate machine learning models capable of accurate predictions.
    - Deploy these models for real-time or batch predictions on new data.

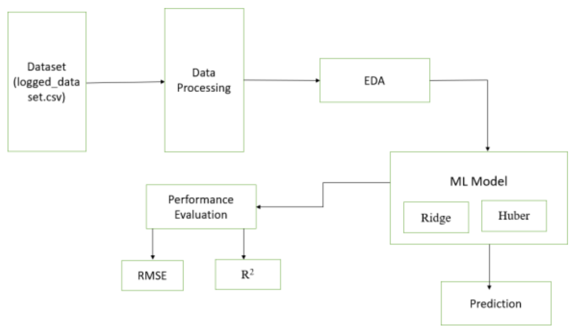

Figure 4.1: Proposed system Block Diagram

Workflow

1. Data Collection

- **Source**: IoT sensors installed in elevator components.
- **Data**: Includes features like vibration (and potentially other metrics such as temperature, pressure, etc.).
- **Target**: revolutions (the performance metric to predict).

2. Data Preprocessing

- **Handling Missing Values**:
    - Replace missing vibration values with the mean of the column.
- **Feature Selection**:
    - Separate the target variable (revolutions) from the features.

- **Data Splitting**:
- Split the dataset into training (80%) and testing (20%) subsets.

3. Exploratory Data Analysis (EDA):

- Visualize relationships between features and the target variable.
- Analyze patterns, distributions, and correlations.

4. Model Selection and Training

- Train multiple regression models to predict revolutions:
  1. **Ridge Regression**: A regularized linear regression model that prevents overfitting by adding an L2 penalty, improving generalization in predictive maintenance.
  2. **Huber Regressor**: A robust regression model that is less sensitive to outliers, making it suitable for handling noisy sensor data in predictive maintenance.

Save trained models to disk for future use.

5. Model Evaluation

- Evaluate each model on test data using metrics:
  - Mean Absolute Error (MAE).
  - Mean Squared Error (MSE).
  - Root Mean Squared Error (RMSE).
  - R^2 Score.
- Visualize model performance with scatter plots comparing actual vs. predicted values.

6. Prediction

- Load test data (new unseen data) and preprocess it.
- Use the trained models to predict revolutions for the test dataset.
- Append the predicted values to the test dataset for further analysis.

7. Deployment and Integration

- Store trained models for deployment in real-world systems.
- Use predictions to optimize maintenance schedules and minimize downtime.

Key Benefits

- **Proactive Maintenance**: Reduces the risk of unexpected failures.
- **Cost Savings**: Lowers maintenance costs by addressing issues early.
- **Improved Safety**: Minimizes risks associated with equipment breakdowns.
- **Operational Efficiency**: Ensures smooth operation of elevators with minimal disruptions.

- Include additional features such as temperature, load, or operational duration.
- Use advanced models like Gradient Boosting or Neural Networks.
- Integrate real-time data streams for dynamic predictions.

Model Building

What is Huber Regressor?

Huber Regressor is a robust regression model that combines the strengths of both linear regression and outlier-resistant techniques. It is particularly useful when dealing with datasets that contain outliers or noisy data.

- Unlike ordinary least squares (OLS) regression, which minimizes squared errors and is highly sensitive to outliers, Huber Regressor minimizes a combination of squared and absolute errors based on a threshold value (delta).
- For small errors, it behaves like linear regression (minimizing squared loss).
- For large errors (outliers), it behaves like absolute error loss (reducing sensitivity to extreme values).
- It is implemented in sklearn.linear_model.HuberRegressor.

Advantages of Huber Regressor

1. **Robust to Outliers**
   - Unlike standard linear regression, which is heavily influenced by extreme values, Huber Regressor reduces their impact.
2. **Balances Efficiency & Accuracy**
   - It provides a good trade-off between mean squared error (MSE) and mean absolute error (MAE), ensuring more stable predictions.
3. **Works Well with Noisy Sensor Data**
   - Ideal for IoT-based predictive maintenance, where sensor readings might contain fluctuations or anomalies.
4. **Improves Generalization**
   - Regularization helps prevent overfitting, making the model more adaptable to unseen data.
5. **Faster than Other Robust Models**
   - Compared to methods like RANSAC or Theil-Senestimators,HuberRegression is computationally efficient.

How Huber Regressor is Used in Our Project?

In our IoT-based predictive maintenance system for elevators, Huber Regressor is used to predict the number of revolutions based on sensor data (vibration, temperature, current, etc.).

- **Why Huber Regression?**
  - Elevator sensor data can have spikes or anomalies due to sudden load changes, mechanical faults, or environmental conditions.
  - Huber Regressor helps in making stable and accurate predictions by reducing the influence of extreme sensor readings.

- **Implementation in Our Project**
    - It takes the training features (X_train) and the target variable (y_train) and fits a regression model that is resilient to outliers.
    - Predictions are made on X_test, and metrics like MAE, MSE, RMSE, and R^2 score are used to evaluate performance.
    - It helps in determining when an elevator might require maintenance, ensuring proactive fault detection.

4. EXPERIMENTAL ANALYSIS

The project is a well-structured implementation of a predictive maintenance system. It demonstrates how to process data, train machine learning models, evaluate their performance, and make predictions. Below is a detailed breakdown of each section of the code:

1. Library Import and Setup

- **Purpose:** Import necessary libraries for data handling, machine learning, and visualization.
- **Key Libraries:**
    - pandas and numpy: Handle data processing and numerical operations.
    - matplotlib and seaborn: For data visualization and plotting.
    - joblib: Save and load machine learning models to/from files.
    - sklearn: Provides tools for model development and performance evaluation.
- **Warnings:** Suppressed to ensure clean output.

2. Data Loading and Preprocessing

- **Loading Data:** Reads the dataset (predictive-maintenance-dataset.csv) into a Pandas DataFrame.
- **Basic Operations:**
    - Display the first and last few rows (head() and tail()).
    - Check for missing values (isnull().sum()) and fill them (using the mean of the vibration column).
    - Check for duplicate rows and remove them if found.
    - Inspect the dataset's structure info() and summary statistics describe().

3. Data Splitting

- **Purpose:** Split data into training and testing sets using train_test_split:
    - X_train and y_train: Used to train the model.
    - X_test and y_test: Used to evaluate the model.
- The test size is set to 20%, and a random seed (random_state=42) ensures reproducibility.

4. Regression Metrics Function

- **Purpose:** Evaluate the performance of regression models using several metrics:
    - **Mean Absolute Error (MAE):** Average absolute difference between predicted and actual values.
    - **Mean Squared Error (MSE):** Average squared difference between predicted and actual values.
    - **Root Mean Squared Error (RMSE):** Square root of MSE, easier to interpret as it's in the same unit as the target.
    - **R^2 Score:** Indicates how much of the target variance the model explains (ranges from 0 to 1).
- **Visualization:**
    - Scatter plot of actual vs. predicted values.
    - A diagonal line represents perfect predictions.

5. Model Training and Evaluation

The code implements two regression models to predict elevator revolutions based on sensor data:

Huber Regressor

- **Checks if a pre-trained model (huber_regressor.pkl) exists:**
    - If yes, loads the model and evaluates it.
    - If not, trains a new Huber Regressor, saves it, and evaluates its performance.
- **Robustness:**
    - Huber Regressor is robust to outliers and handles noisy sensor data well.
- **Predictions and Performance Evaluation:**
    - Uses the defined regression metrics function to assess accuracy.

Result Description

	ID	revolutions	humidity	vibration	x1	x2	x3	x4	x5
0	1	93.744	73.999	18.00	167.743	19.745	1.266828	8787.937536	5475.852001
1	2	93.740	73.999	18.00	167.739	19.741	1.266774	8787.187600	5475.852001
2	3	93.736	73.998	18.00	167.734	19.738	1.266737	8786.437696	5475.704004
3	4	93.732	73.998	18.00	167.730	19.734	1.266683	8785.687824	5475.704004
4	5	93.729	73.998	18.00	167.727	19.731	1.266642	8785.125441	5475.704004
...	...	...	...	...	...	...	...	...	...
111996	111997	20.186	73.992	2.00	94.178	-53.806	0.272813	407.474596	5474.816064
111997	111998	20.186	73.992	20.01	94.178	-53.806	0.272813	407.474596	5474.816064
111998	111999	20.185	73.992	20.01	94.177	-53.807	0.272800	407.434225	5474.816064
111999	112000	20.185	73.992	20.00	94.177	-53.807	0.272800	407.434225	5474.816064
112000	112001	20.184	73.992	2.00	94.176	-53.808	0.272786	407.393856	5474.816064

112001 rows × 9 columns

Figure : Sample Dataset

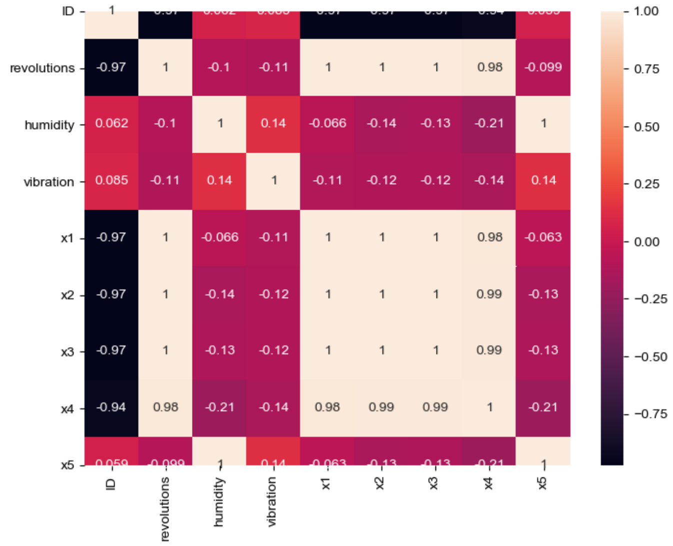

Figure: Heat map for column importance

```
Huber Regressor Mean Absolute Error (MAE): 0.08
Huber Regressor Mean Squared Error (MSE): 0.01
Huber Regressor Root Mean Squared Error (RMSE): 0.10
Huber Regressor R2 Score: 1.00
```

Figure 7.3.3: Displaying the regression report of Huber model

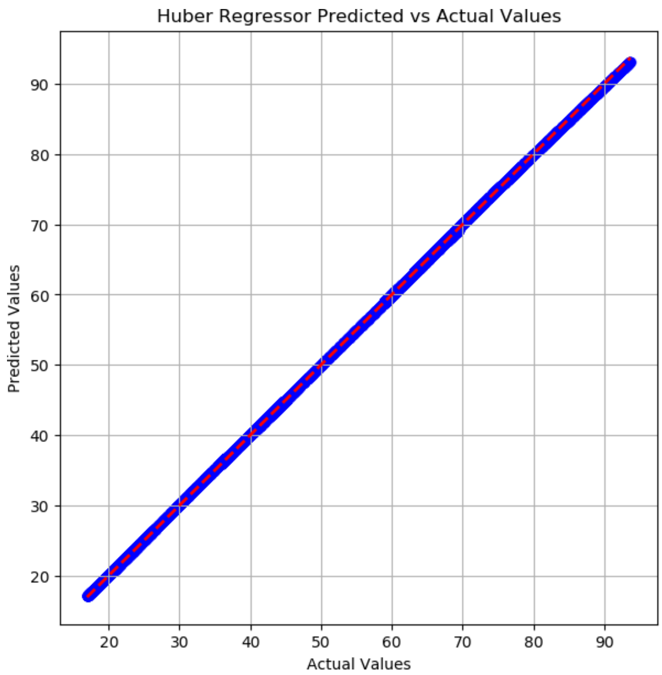

Figure : Illustration of confusion matrix obtained using Huber model.

5. CONCLUSION

The predictive maintenance system for elevators, leveraging IoT sensor data and machine learning models, marks a significant advancement over traditional maintenance approaches. By analyzing critical parameters such as vibrations and revolutions, this system enables early detection of potential failures, transitioning from reactive and scheduled maintenance to a proactive, data-driven approach. The implementation of models like Huber Regressor has demonstrated the capability to provide accurate predictions, reducing unplanned downtime, enhancing safety, and optimizing operational costs.

This project highlights the importance of integrating IoT and machine learning technologies into maintenance workflows. The adoption of predictive maintenance not only extends the lifespan of elevator components but also ensures their consistent performance in high-demand environments. Additionally, the system aligns with global sustainability goals by reducing energy consumption, minimizing material waste, and improving resource efficiency.The scalability and flexibility of the proposed solution allow its application across various industries, including manufacturing, transportation, and healthcare, making it a valuable contribution to the broader field of smart maintenance systems.

REFERENCES

1. Chen, X.; Han, T. Disruptive technology forecasting based on Gartner hype cycle. In Proceedings of the 2019 IEEE Technology & Engineering Management Conference (TEMSCON), Atlanta, GA, USA, 12–14 June 2019; IEEE: Piscataway Township, NJ, USA, 2019.
2. Fessenmayr, F.; Benfer, M.; Gartner, P.; Lanza, G. Selection of traceability-based, automated decision-making methods in global production networks. *Procedia CIRP* **2022**, *107*, 1349–1354.
3. Rajagopal, N.K.; Qureshi, N.I.; Durga, S.; Ramirez-Asis, E.H.; Huerta-Soto, R.M.; Gupta, S.K.; Deepak, S.; Ahmad, M. Future of Business Culture: An Artificial Intelligence-Driven Digital Framework for Organization Decision-Making Process. *Complexity* **2022**, *2022*, 7796507.
4. Mei, J.; Zheng, G.; Zhu, L. Governance mechanisms implementation in the evolution of digital platforms: A case study of the Internet of Things platform. *RD Manag.* **2022**, *52*, 498–516.
5. Umair, M.; Cheema, M.A.; Cheema, O.; Li, H.; Lu, H. Impact of COVID-19 on IoT Adoption in Healthcare, Smart Homes, Smart Buildings, Smart Cities, Transportation and Industrial IoT. *Sensors* **2021**, *21*, 3838.
6. Neves da Rocha, F.; Pollock, N. Innovating in digital platforms: An integrative approach. In Proceedings of the 21st International Conference on Enterprise Information Systems, Heraklion, (ICEIS 2019), Crete, Greece, 3–5 May 2019; ICEIS: Crete, Greece, 2019; Volume 2, pp. 505–515.
7. Tao, F.; Xiao, B.; Qi, Q.; Cheng, J.; Ji, P. Digital twin modeling. *J. Manuf. Syst.* **2022**, *64*, 372–389.
8. Sánchez, R.V.; Siguencia, J.F.; Villacís, M.; Cabrera, D.; Cerrada, M.; Heredia, F. Combining Design Thinking and Agile to Implement Condition Monitoring System: A Case Study on Paper Press Bearings. *IFAC Papers OnLine* **2022**, *55*, 187–192.
9. Ingemarsdotter, E.; Kambanou, M.L.; Jamsin, E.; Sakao, T.; Balkenende, R. Challenges and Solutions in condition-based maintenance implementation—A multiple case study. *J. Clean. Prod.* **2021**, *296*, 126420.
10. Nata, C.; Laurence; Hartono, N.; Cahyadi, L. Implementation of Condition-based and Predictive-based Maintenance using Vibration Analysis. In Proceedings of the 2021 4th International Conference of Computer and Informatics Engineering (IC2IE), Depok, Indonesia, 14–15 September 2021; IEEE: Piscataway Township, NJ, USA, 2021; pp. 90–95.