Journal of Science and Technology
ISSN: 2456-5660 Volume 9, Issue 1 (JAN -2024)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2024.v9.i1.pp30-38*

Real-Time Anomaly Detection in Metro Train APU Compressors: Insights from Operational Data

B. Srinivasulu[1], Sk. Seema[2], R. Jyosthana vineela[2], Sk. Parveen begum[2], Sk. Rizwana[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science Engineering
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India

To Cite this Article

. B. Srinivasulu, Sk. Seema, R. Jyosthana vineela, Sk. Parveen begum, Sk. Rizwana **, "Real-Time Anomaly Detection in Metro Train APU Compressors: Insights from Operational Data"** *Journal of Science and Technology, Vol. 09, Issue 01 - JAN 2024, pp30-38*

Article Info

Received: 25-12-2023 Revised: 05 -01-2024 Accepted: 15-01-2024 Published: 25-01-2024

ABSTRACT

Metro train systems are vital components of modern urban transportation networks. Ensuring the reliable operation of auxiliary power units (APU) is crucial for the overall performance and safety of metro trains. Anomaly detection in APU compressors can help prevent failures and minimize downtime, enhancing the efficiency and reliability of metro services. Conventional methods of anomaly detection in industrial settings often rely on rule-based systems or threshold-based alarms. While these approaches may be effective to some extent, they may not capture subtle anomalies or adapt well to evolving operating conditions. The primary challenge is to develop a system capable of continuously monitoring APU compressors and detecting anomalies in their operation. This involves analyzing operational data in real-time to identify deviations from normal behavior that may indicate impending failures or performance issues. Therefore, the Metro systems are relied upon by millions of commuters daily for efficient and timely transportation. APU compressors play a critical role in maintaining optimal conditions within train compartments. Detecting anomalies in real-time can prevent potential malfunctions or breakdowns, ensuring passenger safety and minimizing disruptions to metro services.The project, "Real-Time Anomaly Detection in Metro Train APU Compressors: Insights from Operational Data," aims to revolutionize maintenance practices in metro systems by leveraging advanced data analytics and machine learning techniques. By collecting and analyzing real-time operational data from APU compressors, this research endeavors to develop a system capable of autonomously and accurately detecting anomalies. The integration of machine learning algorithms allows for the identification of complex patterns indicative of potential issues, enabling timely interventions to prevent failures and ensure the uninterrupted operation of metro train systems. This advancement holds great promise for enhancing the safety, efficiency, and reliability of urban transportation networks.

Keywords: Insights Operational data, Real Time anolomy Detection, Metro train APU Compressors.**1. 1. INTRODUCTION**

Transportation vehicles can be maintained using many strategies, such as preventative maintenance, corrective maintenance, and condition-based maintenance. Preventive maintenance involves conducting frequent inspections according to a predetermined schedule, during which equipment is either replaced or repaired. This form of maintenance results in inefficient allocation of resources towards repairing or replacing equipment that is still functional in order to prevent unexpected breakdowns, as well as the loss of time spent addressing emergencies and analyzing issues. Corrective maintenance [2] involves waiting for a breakdown to happen before repairing the equipment. Condition-based maintenance is a maintenance method that assesses the current state of a system in order to determine the necessary maintenance tasks. Predictive maintenance (PdM) is a technique that relies on data analysis tools to evaluate historical and real-time data from different components of a system. Its purpose is to identify abnormalities and potential equipment flaws, allowing for timely repairs to prevent system failures. In recent years, machine learning techniques, particularly deep learning, have been proposed for predictive maintenance (PdM). Deep learning technologies, such as Deep Neural Network, Recurrent Neural Network, Convolution Neural Network, and Long Short Term Memory, can anticipate the likelihood of equipment failure by autonomously analyzing historical data of the system. Sparse autoencoders are highly effective deep neural networks that have been effectively utilized for the purpose of failure detection. Autoencoders, as a type of machine learning model that does not require labeled data, have the ability to autonomously acquire features from data that does not have predefined categories. A recent analysis of literature reveals that PdM methods can be categorized into three primary classifications: model-based, knowledge-based, and data-driven approaches. Data-driven predictive maintenance (PdM) methodologies [4] identify faults and irregularities by examining the data obtained from various sensors in real-time. The data-driven algorithms efficiently integrate a substantial volume of real-time data from sensors to forecast and identify failures. These algorithms have garnered significant interest in contemporary industrial systems, as evidenced by references [5]–[8]. This research presents the implementation of a data-driven Prognostics and Health Management (PdM) framework using a sparse autoencoder. The framework aims to detect and predict faults in the air production unit (APU) system of a train in Metro do Porto. This system is essential and in high demand for the vehicle's operation. Its breakdown, without any backup, leads to the immediate need for vehicle repair. This has a significant impact not only on the running corporation, but primarily on the customers who witness their expectations of trust in transportation being undermined. The objective is to detect and differentiate between typical and atypical patterns in the data flow derived from a collection of sensors integrated into the APU system during train operation. The goal is to utilize unsupervised methods grounded in deep learning to forecast the progression of a failure.

2. LITERATURE SURVEY

Failures can be detected by finding patterns in data that do not correspond to normal system behavior and which represent anomalies. In the past decades, several anomaly detection approaches have been proposed for of failure prediction or early failure detection, e.g., [9], [10]. More specifically, and regarding railway industry, two recent literature surveys on the work related to different PdM methods can be found in [11] and [12]. Among all the proposed PdM methods for the railway industry, we are interested in data-driven based on learning methods. A recent work in [13] explores data-driven PdM based on anomaly and novelty detection implemented to predict failure in the automatic door system of the train and prevents the spread of breakdown in the system. The authors developed and implemented four common learning algorithms for anomaly detection. Moreover, the results show that a low-pass filter can significantly reduce the number of false alarms. Lee [14] used a logistic regression classifier to model the compressor behavior used for air leakage detection by anomaly detection in a train's braking pipes. Also, a density-based clustering method with a dynamic density threshold was used to distinguish anomalies from outliers and detect anomalies based on the severity degree. More recently, Chen et al. [15] focus specifically on predicting compressor failures using a recurrent neural network using Long Short-Term Memory (LSTM) architecture. The authors compared their method and a random forest method where the experimental results show that predictions by LSTM stay significantly more stable over time, while in terms of AUC score random forest slightly outperforms the LSTM. Most recently, Barros et al. [16] developed a real-time data analysis of the sensors installed on APUs that detects anomalies. They also provided rules based on peak frequency analysis. They considered definition of normal and abnormal behavior of sensors data which can be used for APU failure detection.

3. PROPOSED SYSTEM

Overview

In response to these challenges. The essence of the AI-driven approach involves training these models on meticulously labeled datasets containing examples of different surfaces. Through this training process, the models can autonomously learn to extract relevant features from sensor data, enabling the robot to discern and classify surfaces with heightened accuracy.

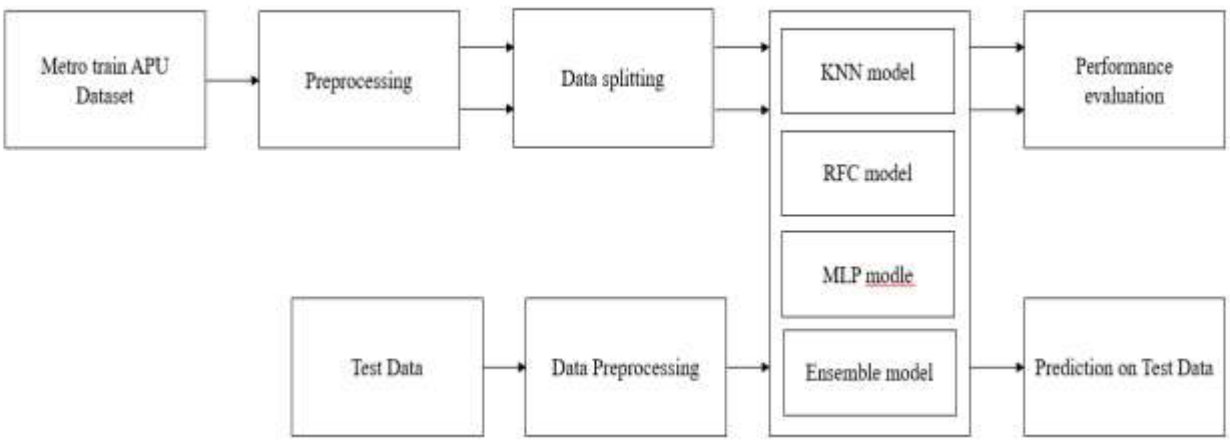

Figure .1: Architectural block diagram of proposed system.

The provided Python script implements a graphical user interface (GUI) application using Tkinter for a surface identification project based on robot-sensed data. Here's a detailed explanation of the steps carried out by the application:

Dataset Upload: The application starts with a button labeled "Upload Dataset." When clicked, this button opens a file dialog, allowing the user to select the dataset file (assumed to be in CSV format). The chosen file is then loaded into the application, and its name is displayed in the text widget. The dataset is stored in the 'dataset' variable.

Dataset Preprocessing: The "Preprocess Dataset" button triggers the preprocessing phase. Missing values in the dataset are filled with zeros, and an overview of the dataset, including the first few records, is displayed in the text widget. Additionally, a count plot is generated to visualize the distribution of classes in the 'surface' column. Label encoding is applied to convert categorical class labels into numerical values.

Train-Test Splitting: The dataset is split into training and testing sets using the scikit-learn train_test_split function. Information about the total number of records in the dataset, as well as the training and testing sets, is displayed in the text widget.

Decision Tree Classifier: The "Decision Tree Classifier" button initiates the training of a Decision Tree classifier. The model is fitted on the training set, and predictions are made on the testing set. The evaluation metrics, including accuracy, confusion matrix, and classification report, are displayed. Additionally, a Receiver Operating Characteristic (ROC) graph is generated to visualize the model's performance.

Random Forest Classifier: The "Random Forest Classifier" button triggers the training of a Random Forest classifier. Similar to the Decision Tree model, evaluation metrics and a ROC graph are displayed in the text widget.

Prediction on Test Data: The "Prediction" button allows the user to select a file for making predictions using the trained Decision Tree classifier. Predictions are displayed in the text widget, indicating the predicted classes for each test data entry.

Performance Estimation and Comparison: The "Comparison Graph" button generates a bar graph comparing performance metrics (precision, recall, F1-score, and accuracy) between the Decision Tree classifier and the Random Forest classifier. This visual representation provides an easy comparison of the two models.

Exit: The "Exit" button closes the Tkinter GUI application.

Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Journal of Science and Technology
ISSN: 2456-5660 Volume 9, Issue 1 (JAN -2024)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2024.v9.i1.pp30-38*

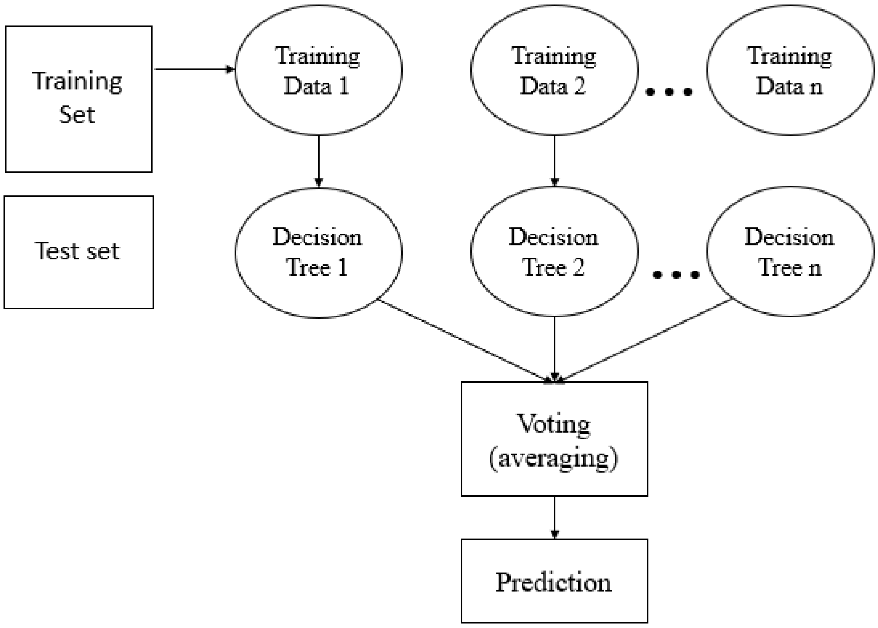

Fig.2: Random Forest algorithm.

Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

4. RESULTS AND DISCUSSION

Dataset Description

the dataset is system and its process where various measurements are recorded over time.

Here's a detailed description of each column:

- timestamp: This column represents the time at which the measurements were recorded. It serves as a chronological reference for the dataset.
- TP2 and TP3: These columns represent measurements or parameters labeled as "TP2" and "TP3." The specific meaning of these parameters would depend on the context of the system.
- H1: This column represents a measurement or parameter labeled as "H1." The specific meaning of this parameter would depend on the context of the system.
- DV_pressure: This column represents a measurement of pressure, related to a device labeled "DV."

Results Description

- The figure 3 represents the graphical user interface (GUI) of the metro train Automatic Power Unit (APU) application. It has different components and options for interacting with the APU system.

ISSN: 2456-5660 Volume 9, Issue 1 (JAN -2024)
www.jst.org.in DOI:https://doi.org/10.46243/jst.2024.v9.i1.pp30-38

- The figure 4 illustrates the process of uploading a dataset to the metro train APU application. It show a file upload dialog or a message confirming the successful dataset upload.
- The figure 5 is associated with the training process of an ensemble model, which could involve combining predictions from multiple models. The figure showcases the ensemble model's performance metrics.
- The figure 6 displays the predicted outcomes on the test data using the trained models. It provides insights into how well the models generalize to new, unseen data.

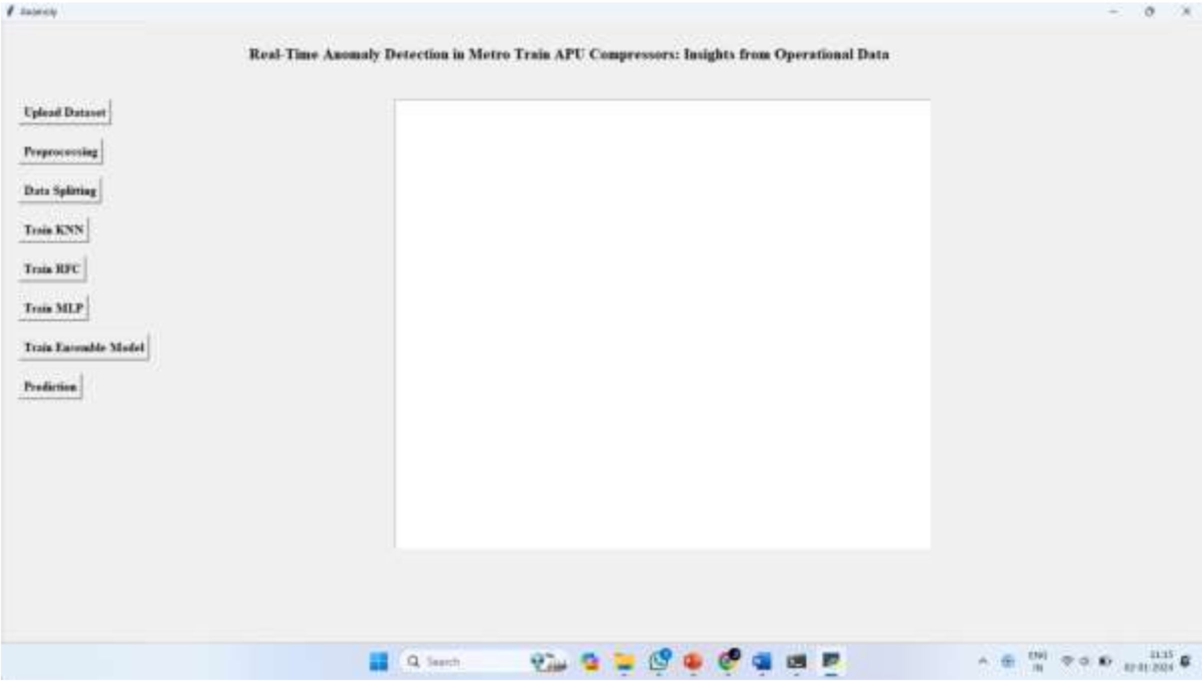

Figure 3: Displays the GUI of metro train APU.

Figure 4: Shows the uploading of dataset.

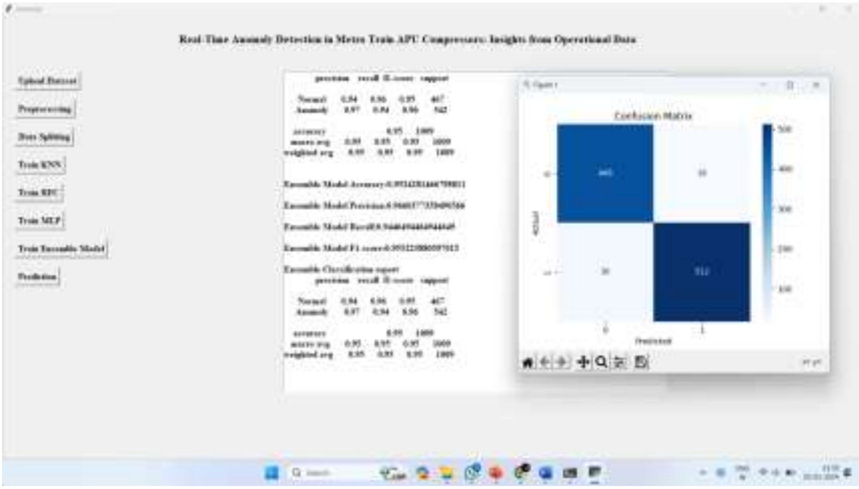

Figure 5: Displays the Ensemble model building and its performance metrices.

Figure 6: Represents the prediction on test data by the model.

5. CONCLUSION AND FUTURE SCOPE

The project on real-time anomaly detection in metro train APU compressors represents a significant advancement in maintenance practices for metro systems. The reliable operation of auxiliary power units (APU) is crucial for the overall performance and safety of metro trains, making the detection of anomalies in APU compressors a critical aspect of ensuring efficient and uninterrupted metro services. The conventional methods of anomaly detection in industrial settings have limitations in capturing subtle deviations and adapting to evolving operating conditions. This research addresses these challenges by leveraging advanced data analytics and machine learning techniques for real-time anomaly detection.By collecting and analyzing real-time operational data from APU compressors, the project aims to develop a system that can autonomously and accurately detect anomalies. The integration of machine learning algorithms enhances the capability to identify complex patterns indicative of potential issues, enabling timely interventions to prevent failures and minimize downtime. This innovation holds great promise for revolutionizing maintenance practices in metro systems, contributing to the safety, efficiency, and reliability of urban transportation networks..

REFERENCES

[1] G. Budai, D. Huisman, and R. Dekker, “Scheduling preventive railway maintenance activities,” Journal of the Operational Research Society, vol. 57, no. 9, pp. 1035–1044, 2006.

[2] C. Stenstrom, P. Norrbin, A. Parida, and U. Kumar, “Preventive and ¨ corrective maintenance–cost comparison and cost–benefit analysis,” Structure and Infrastructure Engineering, vol. 12, no. 5, pp. 603–617, 2016.

[3] J. Yan, Y. Meng, L. Lu, and L. Li, “Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for pre dictive maintenance,” IEEE Access, vol. 5, pp. 23 484–23 491, 2017.

[4] W. Zhang, D. Yang, and H. Wang, “Data-driven methods for predictive maintenance of industrial equipment: a survey,” IEEE Systems Journal, vol. 13, no. 3, pp. 2213–2227, 2019.

[5] K. Wang, “Intelligent predictive maintenance (ipdm) system–industry 4.0 scenario,” WIT Transactions on Engineering Sciences, vol. 113, pp. 259–268, 2016.

[6] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring,” Mechanical Systems and Signal Processing, vol. 115, pp. 213–237, 2019.

[7] G. Li, C. Deng, J. Wu, X. Xu, X. Shao, and Y. Wang, “Sensor data-driven bearing fault diagnosis based on deep convolutional neural networks and s-transform,” Sensors, vol. 19, no. 12, p. 2750, 2019.

[8] Y. Jiang and S. Yin, “Recursive total principle component regression based fault detection and its application to vehicular cyber-physical systems,” IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1415–1423, 2017.

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM computing surveys (CSUR), vol. 41, no. 3, pp. 1–58, 2009.

[10] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” arXiv preprint arXiv:1901.03407, 2019.

[11] E. Fumeo, L. Oneto, and D. Anguita, “Condition based maintenance in railway transportation systems based on big data streaming analysis,” Procedia Computer Science, vol. 53, pp. 437–446, 2015.