Environmental Science

https://www.aimspress.com/journal/environmental

AIMS Environmental Science, 11(6): 960–986
DOI: 10.3934/environsci.2024048
Received: 05 March 2024
Revised: 02 June 2024
Accepted: 19 September 2024
Published: 25 November 2024

Research article

A multi-task model for failure identification and GPS assessment in metro trains

Pratik Vinayak Jadhav[1], Sairam V. A[1], Siddharth Sonkavade[1], Shivali Amit Wagle[1], Preksha Pareek[2], Ketan Kotecha[3,*]and Tanupriya Choudhury[1,4]

[1] Symbiosis Institute of Technology (Pune Campus), Symbiosis International Deemed University, Pune 411215, India

[2] Computer Engineering Department, Thakur College of Engineering and Technology, Mumbai 400101,India

[3] Symbiosis Centre for Applied Artificial Intelligence, Symbiosis Institute of Technology, Symbiosis International Deemed University, Pune 412115, India

[4] School of Computer Sciences, UPES, Dehradun, Uttarakhand 248007, India

* **Correspondence:** Email: head@scaai.siu.edu.in.

Abstract: Air and oil leaks are two of the predominant operational failures in metro trains, which can cause severe issues and a lot of downtime. Predictive maintenance on such machinery can be of great use. This work aimed to develop a deep learning algorithm for fault analysis in metro trains. The MetroPT dataset was used for this work. A multi-task artificial neural network was developed for the simultaneous identification of failures and GPS quality assessment. The network had common dense, batch normalization, and Gaussian noise layers, followed by output sigmoid layers for each output. The algorithm was trained for 20 epochs with a batch size of 5000 using the using Adam optimizer. The local interpretable model agnostic explanations (LIME) technique was used to provide explanations for the model predictions. Finally, a dashboard was developed for the same application consisting of the best-trained algorithm for decision-making, along with trend visualizations and explanations. The developed multi-task model produced 98.89%, 99.12%, and 99.24% accuracies in the testing set for failure type, failure location, and GPS quality predictions, respectively. The model produced 99.56%, 99.67%, and 99.84% precision in the testing set for failure type, failure location, and GPS quality predictions, respectively. The loss values for the trained model on the testing set were 0.0035, 0.0026, and 0.0033 for the three tasks, respectively. The deep learning model took 43 seconds for training and 1 second for inferencing for test data. The LIME technique produced explanations for each predictive task with feature importance in positive and negative impacts. On the whole, the proposed framework can be effective for fast and accurate fault analysis in metro trains.

Keywords: predictive maintenance; multi-task learning; machine learning; deep learning; metro trains; Industry 4.0

1. Introduction

Machinery has become an integral part of human life, especially considering the technological advancements related to Industry 4.0. Failures occurring on such crucial machinery lead to unplanned downtime, ultimately resulting in loss in economic aspects [1]. This is catastrophic in industry and public transport since these failures stop production, causing hassles to the public. Hence, machine diagnostics is of high importance in such instances. Fault prediction in an early stage dramatically improves the machine's lifetime, reducing costs and preventing downtimes.

Air and oil leaks are two of the predominant operational failures in public transport modalities, especially in metro trains, which are our prime objective. Air leaks are prone to occur in the dryer component, whereas oil leaks are prone to occur in the compressor component [2]. Various sensors, like pressure transducers, pneumatic sensors, motor current, etc., are used to analyze and diagnose the faults in metro trains [3–5]. An abnormal change can be observed in these sensors in the event of a fault in that component. Continuous monitoring of the vulnerable components with the sensors mentioned above can help identify the occurrence of a fault in that component [6].

Predictive maintenance has been an emerging technology in machine diagnostics, aiming to predict faults early and perform maintenance to prevent catastrophic events [7]. Also, anomaly detection based on sensor data on an early scale will reduce maintenance expenses and avoid downtime. Data curation, data pre-processing, diagnosis, and decision-making are the critical aspects of predictive maintenance [8]. This data-driven approach has been proven effective due to the vast availability of data and intelligent algorithms for automated analysis [9].

Artificial intelligence plays a promising role in fault prediction and predictive maintenance. Several machines and deep learning algorithms are trained on continuous data from sensors attached to the target machinery [10]. The proposed work uses machine and deep learning algorithms for anomaly detection (air and oil failure) on the air production unit of metro trains and real-time dashboard development, which is the first of this work as per our knowledge. The following section explains the existing works on machine usage and deep learning algorithms for predictive maintenance.

The following are the contributions of the proposed work:

1. To develop a deep learning algorithm for the simultaneous identification of the type and location of the fault, along with GPS quality monitoring from sensor data in metro trains.
2. To integrate an explainable AI technique into the model's prediction and highlight the key sensors contributing to the fault.
3. To develop a dashboard integrating the sensor data analytics as visual graphs, the deep learning model, and the explainable AI results for analysis by engineers.

This work will be of great aid to maintenance engineers for fault analysis in sensor values and assessment of GPS signals in real time. This predictive maintenance application can aid in reducing the downtime and service costs of machine parts, if found damaged.

The following is the outline of the research paper. Section 2 briefs about the existing work relevant to the field of interest. Section 3 explains the material description, the pre-processing techniques used, the different machine learning algorithms, the training parameters, and the methodologies for dashboard development. Section 4 represents the proposed methods' results, graphs, and supporting diagrams. Section 5 compares the proposed work with existing works and draws significant conclusions and future scope.

2. Related works

The detection of air leakage from the pneumatic door of a train is an attempt to reduce train downtime. Deep learning algorithms were applied for automatic feature extraction from extensive data obtained from continuous monitoring by sensors for the task of fault detection [11]. The OSR (open set recognition) concept was used for multi-task classification to predict the known class and detect unknown samples. A lightweight convolutional neural network (CNN) model streamlined with the OSR technique was trained to predict the air leakage. An 8-layer neural network consisting of 6 convolutions and two dense layers was used for air leakage. This model was trained using an SGD (stochastic gradient descent) optimizer with a learning rate of 0.001 for a batch size of 64.

The server air leakage in the breaking pipe results in breaking issues and decreases the train reliability [12]. Due to the visual constraints for air leak detection, the paper proposes a framework for the simultaneous prediction of the type and severity of air leakage using anomaly detection methods based on the on-and-off logs of the compressor. Around 632,683 data points were collected from May 2016 to October 2016 from 178 VIRM trains, of which 6957 are labeled as ”Air Leakage” and 625,726 are labeled as ”Normal”. They have used a logistic classifier model for two different classes of compressor behavior for each separate train. One defines the boundary by separating two classes under everyday situations, and the other models the distribution of the compressor idle time and run time separately using logistic functions. It also further detects the context of compressor idle time erroneously classified as a compressor run time, aiding in anomaly detection. A density-based unsupervised clustering approach is adopted for anomaly detection before four weeks and can pre-filter anomalies to prevent false alarms.

The challenges encountered by traditional manufacturing companies during their transition to intelligent factories, notably the scarcity of historical data for training machine learning models, were addressed by Mohan Rajashekarappa et al. [13]. A novel approach of artificially inducing anomalies for data labeling was introduced, and it underscored the importance of proactive readiness for potential future disruptions in newly installed systems. Through two experiments focused on air leakage detection, the proposed methodology demonstrates exceptional performance with RUS-Boosted bagged trees, yielding 98.73% accuracy, 99.40% precision, recall of 99.21%, and an F1 score of 99.30% on the test data.

The critical issue of energy efficiency and fault detection in air conditioning systems emphasizes their intricate nature and substantial energy consumption impact [14]. The study comprises two essential components: First, it investigates the ramifications of various faults within the air conditioning system on its coefficient of performance (COP), shedding light on the potential energy wastage associated with these faults. The research convincingly demonstrates that different faults lead to varying degradation levels in the COP. Second, the paper evaluates the effectiveness of three supervised learning classifier models in classifying these faults: deep learning, support vector machine (SVM), and multi-layer perceptron (MLP). The research assesses the performance of these classifiers across six distinct fault classes, revealing that different faults indeed exert varying negative impacts on the COP.

Predicting air failure of the air production unit (APU) in metro trains. The dataset used for this task was MetroPT, a 6-month analysis of metro trains in Portugal comprising analog, digital, and GPS sensors [15]. The GPS information was excluded from the dataset, and the timestamp was encoded using the label encoding technique. A random forest classifier algorithm was used for the multi-class classification of air failure prediction. The data was undersampled and then split into training and testing sets. A feature importance visualization technique was employed to identify the root cause of the air failure. The random forest classifier produced 85% and 97% accuracies for the binary and multi-class classification tasks, respectively.

A deep learning neural network for anomaly detection in metro trains was developed by Davari et al. [16]. The algorithms used for this task were the sparse autoencoder and variational autoencoder. This work is an unsupervised learning approach for anomaly detection of air failures in trains. The MetroPT dataset was used for this work with sensors placed in the air production unit. The two versions of the autoencoder were used for sensor data reconstruction, and a low-pass filter was used to perform anomaly detection and detect faults. The autoencoder algorithms using the digital data produced precision, recall, and F1 scores of 44%, 13%, and 32% better than that of the algorithms trained on the analog data.

An expert system for the multi-objective optimization of equipment was developed for highway optimization by Ali et al. [17]. The particle swarm optimization was used to simultaneously optimize the time, cost, and quality of the equipment for construction. This method reduced the time and cost by 35.4% and 39.1%, respectively. The application of predictive maintenance in concrete manufacturing was done by Alshboul et al. [18]. Seven different classification algorithms were used, out of which the cat boost classifier produced an F1-score of 0.985, an accuracy of 0.984, a recall of 0.983, and a ROC curve area of 0.984. A comparative analysis of machine learning algorithms for concrete strength estimation was performed by Alshboul et al. [19]. Three machine learning algorithms, namely, XGBoost, LighGBM, and genetic programming, were used, out of which the LightGBM and XGBoost algorithms surpassed other studied algorithms with a coefficient of determination of 95.74% and 93.27%, respectively.

The proposed work aims to develop a decision process for failure prediction and identification of the failure type and location using machine learning and multi-task models using deep learning algorithms.

3. Materials and methods

3.1. *Proposed workflow*

The proposed work adopts the following workflow consisting of different blocks: data acquisition, data pre-processing, feature pre-processing, visualization, model development, validation, and deployment. Figure 1 represents the proposed workflow visually.

3.2. *Dataset description*

The dataset used for this work is named MetroPT [17, 18], comprising of sensor information related to urban metro trains in Portugal collected during the year 2022. The dataset comprises different analogue, digital, and GPS sensors continuously capturing data from the metro trains for six months. The MetroPT dataset has been curated to develop AI algorithms for automated fault prediction and predictive maintenance of metro trains based on sensor data. Table 1 represents the different sensors, their description, and the unit of measurement used for acquiring the MetroPT dataset.

3.3. *Data pre-processing*

The dataset comprises around 15 million sensor data records from Jan 1, 2022 to Jun 30, 2022. The dataset obtained from the source is not directly labeled. The dataset owners have provided information about the failure, like the start time, end time, type, and location of the failure. Based on the start and end times, the appropriate timestamps were found and the values between those timestamps were coded according to the fault type. Table 2 shows the statistical description of the dataset for the parameters, namely, TP2, TP3, H1, DV_pressure (DVP), Reservoirs, Oil_temperature (OT), Flowmeter, Motor_current (MoC), COMP, DV Electric (DVE), Towers, MPG, LPS, Pressure_switch (PrS),Oil_level (OL), Caudal_impulses (CaI), GPS Longitude (GPSLong), GPS Latitude (GPSLat), GPS Speed, GPS Quality, month, day, hour, minute, second. The statistical values of mean, standard deviation (std), minimum value, 25%, 50%, 75%, and maximum values for a dataset are evaluated here. Table 3 represents the label code for the different types of faults occurring in the metro train.

Timestamp data cannot be processed by machine learning and deep learning algorithms. Hence, it needs to be processed. This issue was tackled by extracting the month, week, day, hour, minute, and second information from the timestamp using the Pandas functionalities. Finally, the timestamp feature is removed from the dataset. Two new columns, one for the type of failure and another for the location of the failure, were created from the labels. These columns are created as these are the labels for the multi-task model. Table 4 represents the different failure types and location codes.

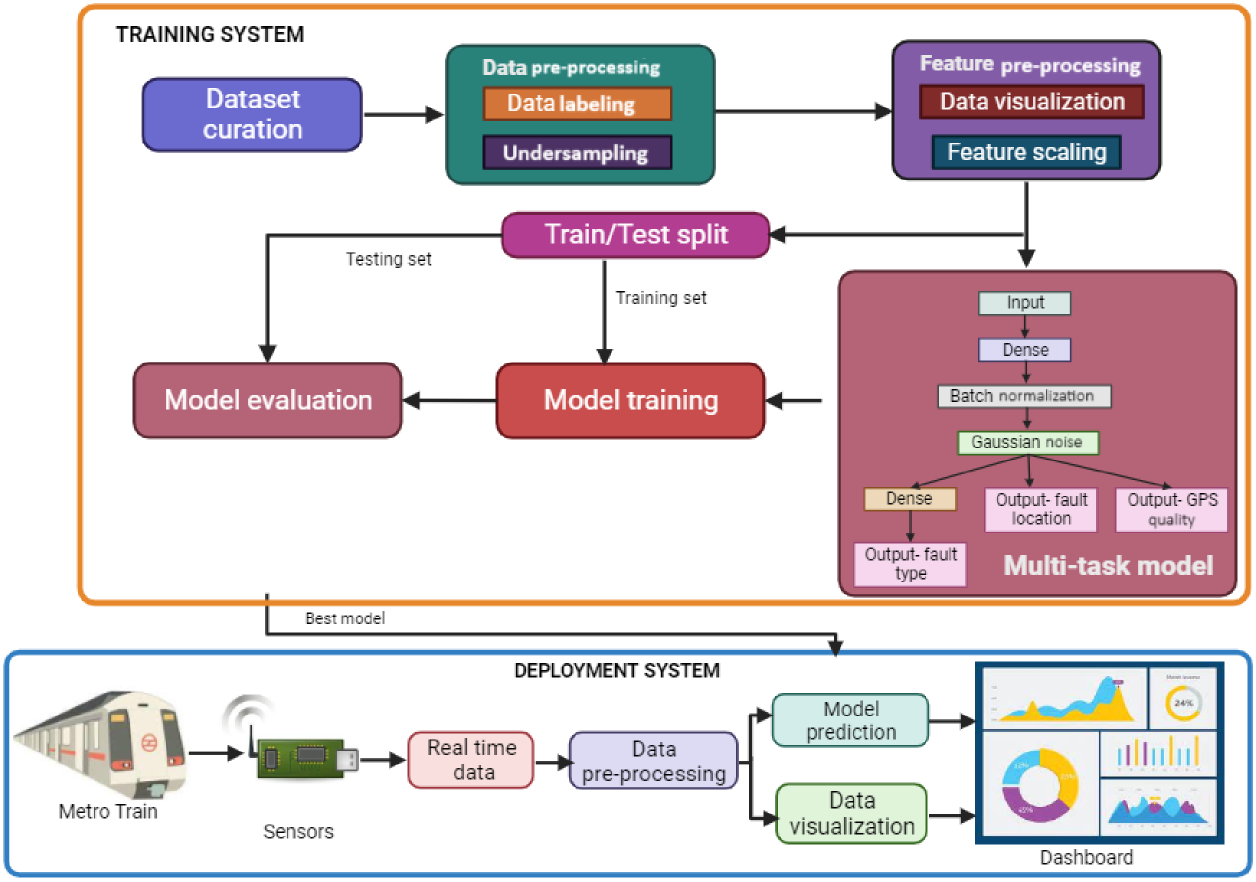


Figure 1. Pictorial representation of the proposed workflow.

The final step in the data pre-processing was the operation of undersampling, especially in the dataset of stage 1. The stage-1 dataset comprises 2,12,104 samples under the fault condition, about 1.63% of the entire data. Around 98.4% of the data belongs to the regular class, proving the dataset is highly imbalanced. Hence, the samples from the regular class were undersampled. Around 28,00,000 randomly selected samples were taken from the entire dataset, which was used as the data for the first stage. The final dataset comprises about 25 input features and three target vectors. Table 5 represents the description of all the targets in the final dataset.

Table 1. Name, description, type, and units of the different sensors used in the metroPT dataset acquisition.

Name	Description	Type of sensor	Unit
TP2	Compressor pressure	Analog	Bar
TP3	Pneumatic panel pressure	Analog	Bar
H1	Pressure of the valve that is activated when the pressure exceeds 10.2 bar	Analog	Bar
DV_Pressure	Pressure drop due to water discharge by air dryers	Analog	Bar
Reservoirs	Air tank pressure	Analog	Bar
Oil temperature	Temperature of oil in compressor	Analog	Celsius
Flowmeter	Airflow	Analog	m3/h
Motor current	Current flowing in the motor	Analog	Ampere
Comp	Electric signal of the compressor based on the air intake	Digital	-
DV Electric	Electric signal of compressor outlet	Digital	-
Towers	Specifies the two towers based on the action of air drying	Digital	-
MPG	Trigger to start the compressor when the pressure is less than 8.2 bar	Digital	-
LPS	Trigger when the pressure is less than 7 bar	Digital	-
The pressure switch	Trigger when pressure is detected in the pilot control valve	Digital	-
Oil level	Trigger when the oil level is less than the threshold	Digital	-
Caudal Impulses	Trigger for the air flowmeter	Digital	-
GPS Longitude	Longitude position	Analog	°
GPS Latitude	Latitude position	Analog	°
GPS Speed	Speed	Analog	Km/h

Table 2. Statistical decription of the dataset.

	TP2	TP3	H1	DVP	Reservoirs	OT	Flowmeter	MoC
mean	0.947	8.989	8.038	-0.019	1.63	65.843	20.128	2.040
std	2.836	0.667	2.846	0.185	0.064	5.931	3.578	2.198
min	-0.03	0.937	-0.033	-0.036	1.349	18.575	18.8347	-0.009
25%	-0.009	8.492	8.332	-0.0279	1.608	61.825	18.9748	0.0024
50%	-0.007	8.996	8.876	-0.025	1.635	66.475	19.03	0.007
75%	-0.006	9.506	9.438	-0.025	1.667	70.575	19.040	3.837
max	10.806	10.38	10.368	8.11	1.791	80.174	37.008	9.3375
	COMP	DVE	Towers	MPG	LPS	PrS	OL	CaI
mean	0.892	0.107	0.946	0.892	0.004	0.0	0.0	0.002
std	0.309	0.309	0.225	0.309	0.068	0.0	0.0	0.047
min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
50%	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
75%	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
max	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0
	GPSLong	GPSLat	GPSSpeed	GPSQuality				
mean	-7.880	37.578	8.592	0.912				
std	2.443	11.649	14.096	0.282				
min	-8.69	0.0	0.0	0.0				
25%	-8.66106	41.1696	0.0	1.0				
50%	-8.658	41.1858	0.0	1.0				
75%	-8.583	41.212	16.0	1.0				
max	0.0	41.240	286.0	1.0				
	month	day	hour	minute	second			
mean	1.0	16.046	13.139	29.507	29.499			
std	0.0	8.919	6.444	17.318	17.318			
min	1.0	1.0	0.0	0.0	0.0			
25%	1.0	8.0	9.0	15.0	14.0			
50%	1.0	16.0	14.0	30.0	29.0			
75%	1.0	24.0	19.0	45.0	44.0			
max	1.0	31.0	23.0	59.0	59.0			

Table 3. Faults occurring in the metro train along with their corresponding label code.

Label Code	Corresponding fault
0	Air leak in air dryer
1	Air leak in client chamber
2	Oil leak in compressor
3	No fault

Table 4. Label code for the type and location of the failure.

Label Code	Type of failure	Location of failure
0	Air leak	Air dryer
1	Oil leak	Client chamber
2	No failure	Compressor
3	Not reported	No location

Table 5. Description and classes for the target vectors in the final processed dataset.

Name of target vector	Description	Classes
Type	Code for the type of failure	0- Air failure 1- Oil failure 2- Normal
Location	Code for the location of the failure	0- Air dryer 1- Client 2- Compressor 3- No location
GPS quality	Quality of the GPS sensor	0- Good 1- Bad

3.4. Feature pre-processing and visualization

The dataset has no null or duplicate values since the dataset is obtained from sensors that continuously monitor the trains. The feature scaling technique, normalization, was adopted to bring all the features to the same scale (0-1). Appropriate data visualization techniques were used for the dataset's univariate, bivariate, and multivariate analysis. The dataset comprises continuous (analogue sensors) and categorical features (digital sensors); we have split them for data visualization. A histogram with a kernel density estimator function is used to visualize the continuous features in the dataset. A pie chart is used to visualize the categorical features. Finally, the information was visualized using a map that represents the train's route along with the train's speed. Figures 2, 4, 3, and 5 represent the visualization plots obtained from the continuous, categorical, GPS, and entire dataset, respectively. After visualization, a stratified split of ratio 80:20 was made for the training and testing sets, respectively.

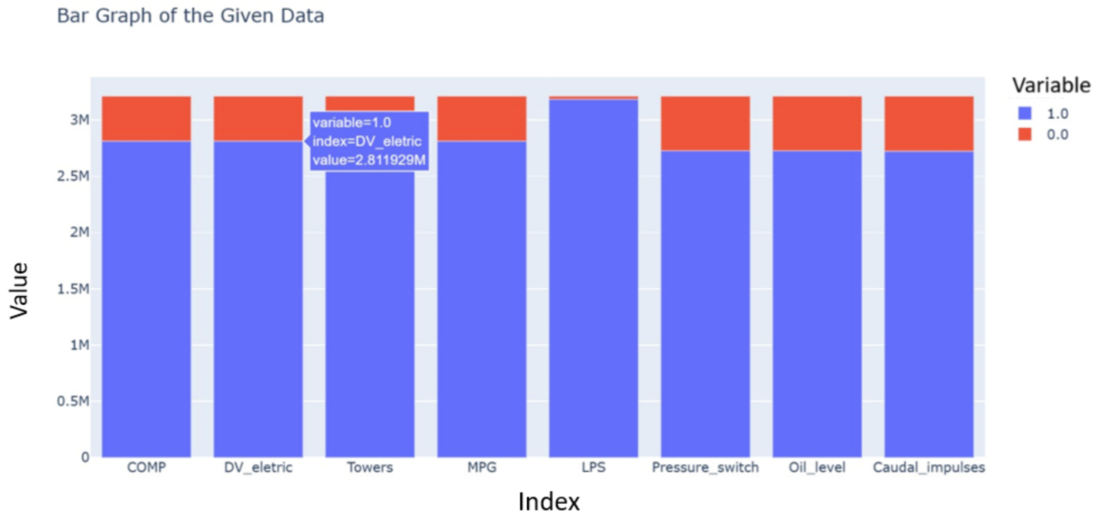


Figure 2. Bar plot for the digital sensors in the dataset.

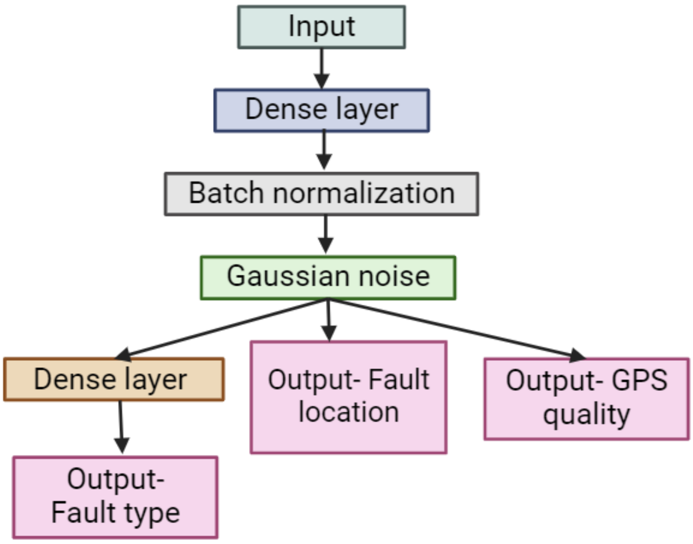


Figure 6. Architecture diagram for the proposed multi-task model.

3.5. Multi-task neural network

Multi-task learning is the ability of a neural network to simultaneously obtain multiple outputs from a single input. In this work, based on the sensor information, the multi-task neural network is designed to simultaneously predict the type and location of the failure in the metro train. The multi-task neural network has common pre-processing layers, followed by branches, each corresponding to a particular task. Hence, the multi-task neural network uses parallel processing to simultaneously identify the type and location of the fault and assess the GPS quality.

The multi-task neural network comprises shared layers and task-specific layers. The input layer, a single hidden layer with four neurons, and regularization layers like dropout and batch normalization are common for both tasks. In contrast, there are individual output layers for each task. The output

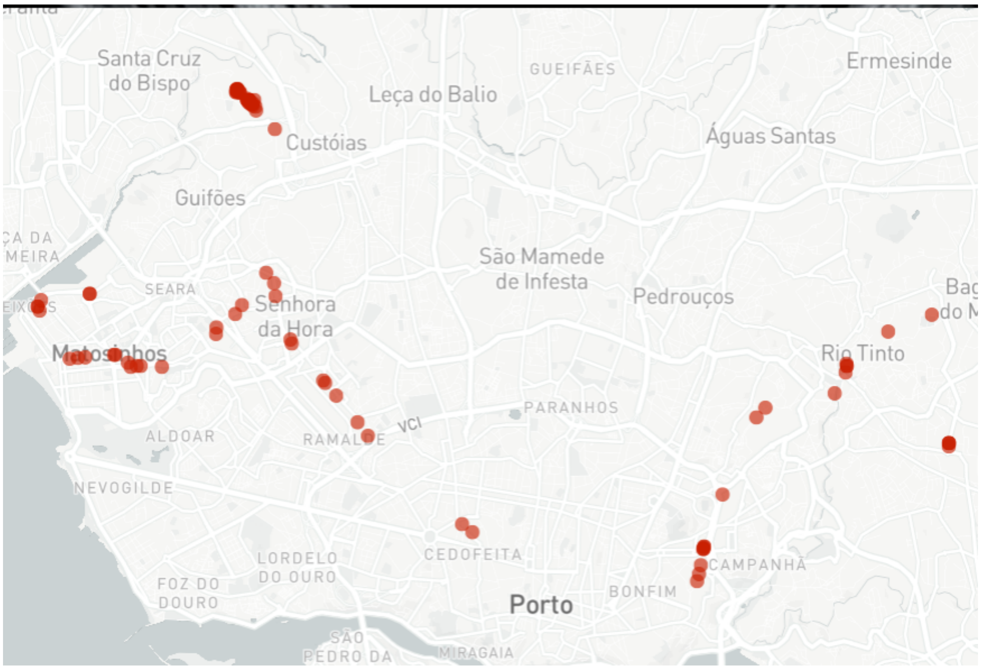


Figure 3. Map visualization of the GPS information of the metro train. The intensity of the red dots represents the train's speed at those points.

layer comprises three neurons for the task of failure identification, one neuron for the task of location identification, and one neuron for the task of GPS quality identification. Figure 6 represents the architecture diagram of the proposed multi-task model.

3.6. *Training methodologies*

The abovementioned multi-task neural networks were trained using the Adam optimizer and a combination of categorical and binary cross-entropy loss functions. The batch size was set to 5000, and the models were trained for 20 epochs. A hybrid loss function was used to train the multi-task model since two tasks (fault type and location identification) were multi-class. In contrast, the task of GPS quality identification is binary. Equation 3.1 represents the hybrid loss function used to train the multi-task model.

$$LF = \sum_{i=1}^{2} \frac{1}{N} \sum_{J}^{n} \sum_{k=1}^{M} -y_{kj} log(p(y_{kj})_i - \frac{1}{N} \sum_{L}^{N} \sum_{p=1}^{2} y_{lp} log P_{lp} \tag{3.1}$$

3.7. *Evaluation metrics*

Evaluation is an essential component of the proposed workflow. Model evaluation is done to identify the model's performance on the testing set. Evaluation of the testing set is essential to identify if the trained model has overfit or underfit. The following performance metrics are derived from the confusion matrix.

The confusion matrix comprises four values: true positive, true negative, false positive, and false negative. The diagonal elements of the matrix indicate the correctly classified samples (true positive and true negative), and the non-diagonal elements indicate the misclassified samples (false positive and false negative). The following are the different metrics used to evaluate the trained models.

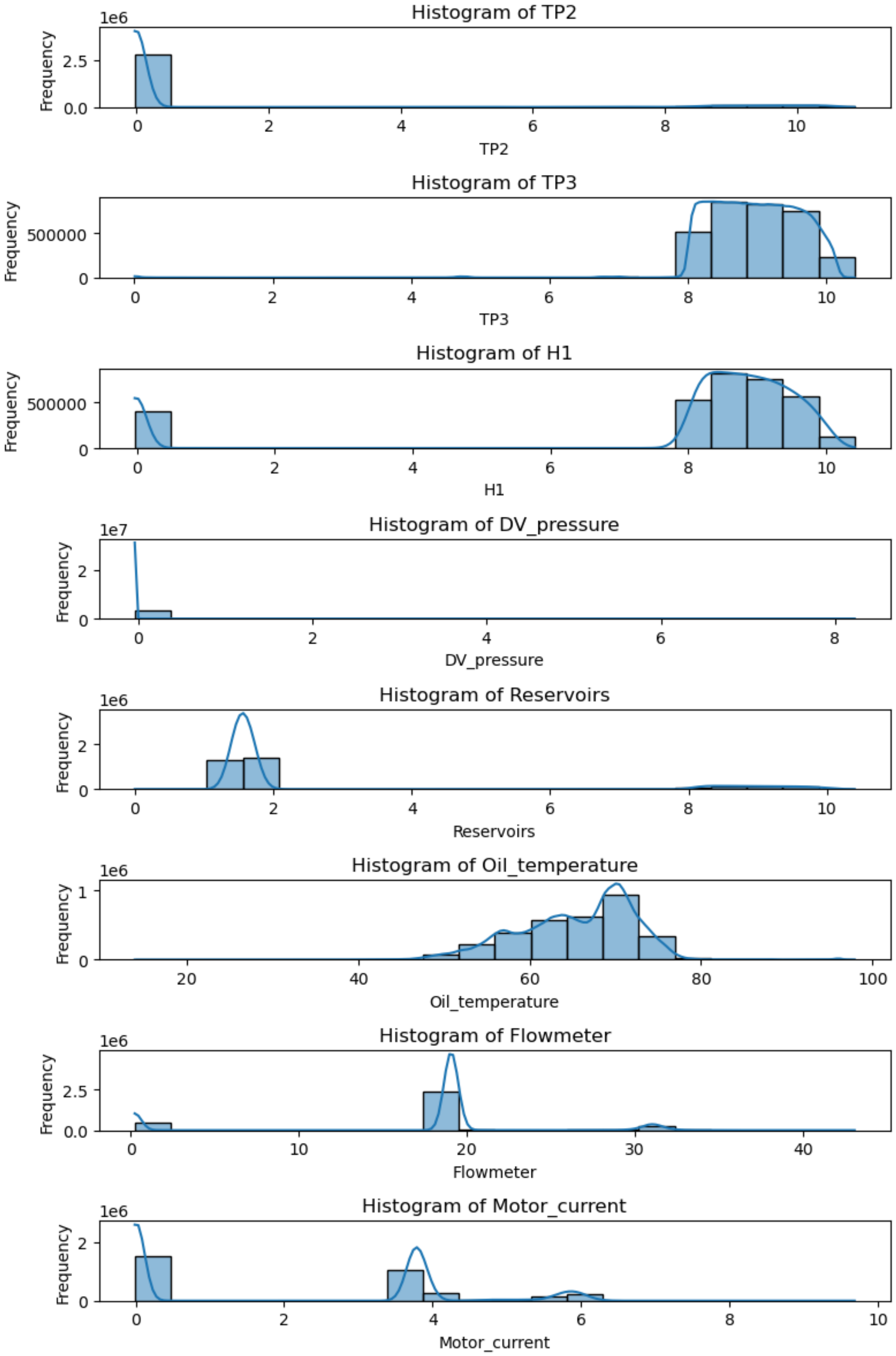


Figure 4. Histogram plot with a kernel density estimator for the analog sensors in the dataset.

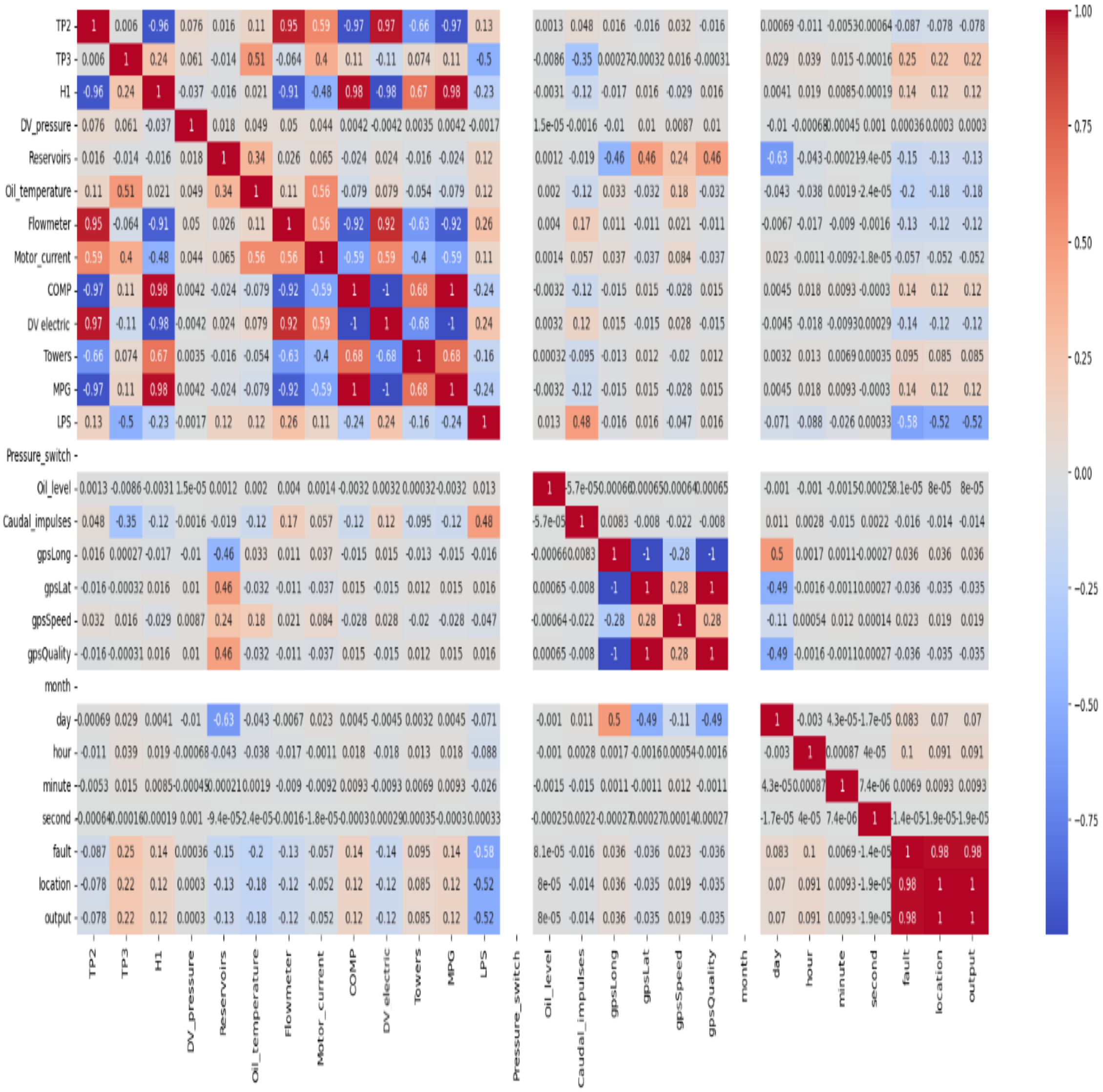


Figure 5. Heatmap visualization for the entire dataset along with the target variables.

3.7.1. Accuracy

Accuracy is defined as the ratio of the correct classifications to that of the total classifications. Accuracy is considered the gold standard metric for the evaluation of classification algorithms. The formula for accuracy is mentioned in Equation 3.2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

3.7.2. Precision

Precision is defined as the ratio of true positives to that of the total positives. Precision is one of the metrics used to analyze a model's performance in class imbalance conditions. The formula for precision is mentioned in Equation 3.3.

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

3.7.3. Recall

Recall is defined as the ratio of true positives to that of total samples. Recall is another metric that is used to analyze multi-class classification under the condition of class imbalance. Equation 3.4 represents the mathematical formula for recall.

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

3.7.4. AUC score

AUC-ROC is expanded as the area under the regional operating characteristic curve. The ROC curve is the plot between the false positive and true positive values. The area under that curve is termed an AUC score. An AUC value of less than 0.5 is considered a terrible score, a score of 0.5 is considered a random guess, and a score of more than 0.8 is considered a good score.

3.8. Interpretability techniques

Model interpretability has been a focus and requirement for which the demand has risen in recent years. Many machine learning and deep learning algorithms are considered black boxes, providing output for input data without any logical interpretations. This is needed in areas like healthcare, where life-concerning critical decisions are made. In such instances, providing interpretability to the model by providing explanations of the predictions can greatly aid clinicians.

In this work, the local interpretable model agnostic explanations (LIME) [19] is used to derive the interpretations of the complex ensemble classifier. As the name suggests, LIME works locally, meaning it works on individual data samples. Also, this method is model agnostic, meaning it works for all models. LIME works by mapping a simple interpretable model (like linear regression) on a complex model [20]. The local region of the data space is considered, where synthetic samples are generated based on original samples. These synthetic samples are labeled based on the prediction of the complex model. Then, a simple interpretable linear regression is trained on the synthetic labeled data. The coefficients of the trained linear regression model represent the interpretations of the complex model on the local space. The LIME tabular function from the LIME library is used to get the interpretations for individual data samples.

Table 6. Performance comparison of the trained multi-task model.

Set	Time consumption per epoch (ms)	Metric	Type identification	Location identification	GPS quality identification
Training	43	Loss	0.0031	0.0020	0.0029
		Accuracy	99.94	99.998	99.99
		Precision	99.99	100	100
		Recall	100	100	100
		AUC	1	1	1
Testing	1	Loss	0.0035	0.0026	0.0033
		Accuracy	98.89	99.12	99.24
		Precision	99.56	99.67	99.84
		Recall	99.92	99.93	99.93
		AUC	1	1	1

The LIME function explains the type of fault, location of the fault, and quality of the GPS sensor, respectively. The tabular explainer from LIME was used to explain the predicted instances. A LIME explainer was used for each task: failure type, failure location, and GPS quality. The LIME explainer generates a figure that shows the input that contributed to that particular class (positive) and the features that contributed to the counter-class (negative). Hence, we can understand the features that positively and negatively contribute to a particular class from the plot.

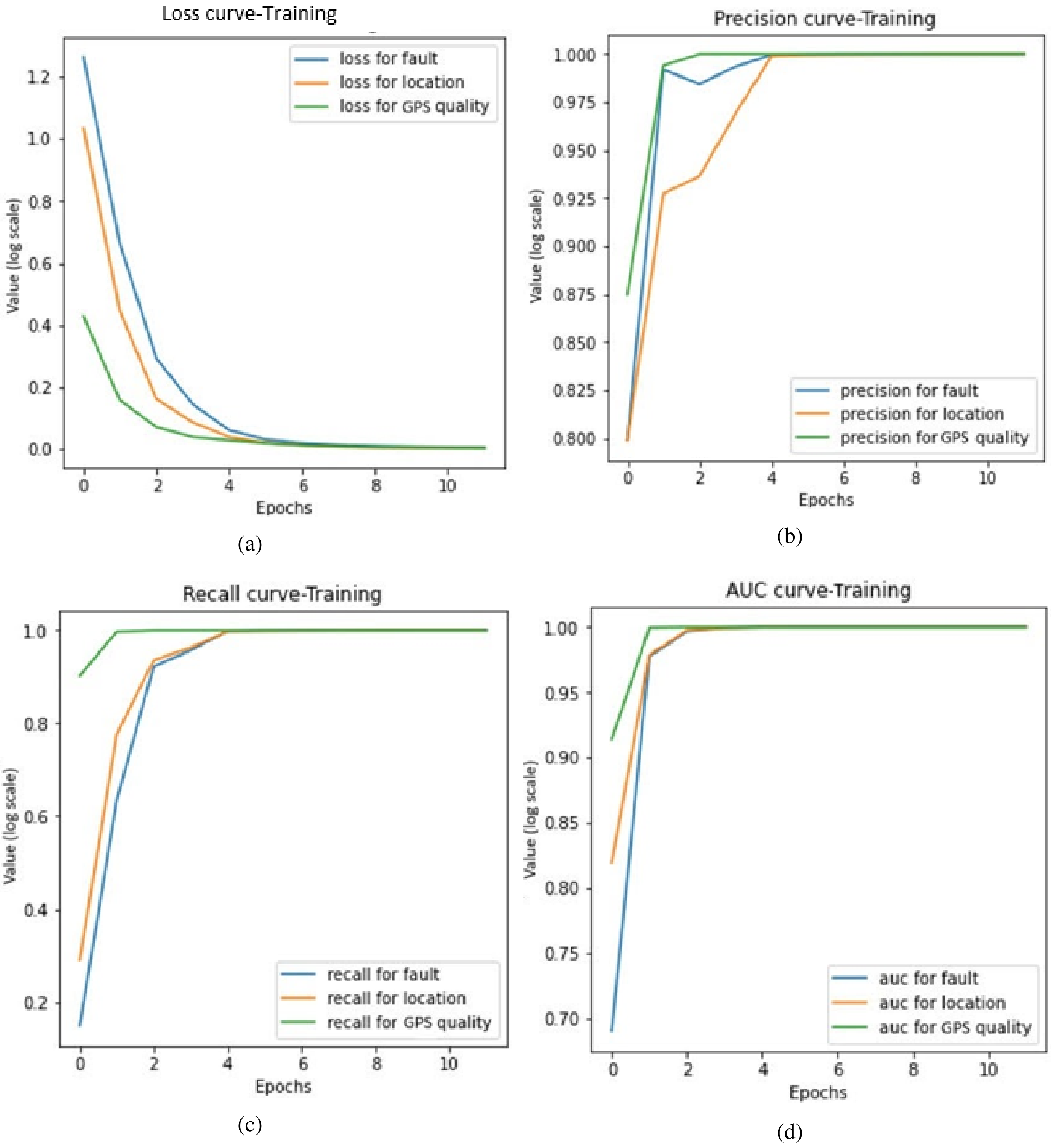
3.9. Dashboard development

The ultimate aim of the work is to develop a dashboard for real-time data analytics and predictions. A website was developed using a Python-based web development tool. The website will request the data recorded from the sensors as an Excel sheet of CSV (comma-separated values). The data visualization and prediction tasks are done simultaneously upon receiving the data. On the data visualization task, a stacked area chart of the continuous features, a stacked bar chart for the categorical features, and a map depicting the speed and route of the train are made. For the data prediction task, the latest values are processed and sent to the trained multi-task model for predictions on the failure type, failure location, and quality of the GPS, which are displayed on the website. Finally, the LIME explanations for the model on the provided sample are given for all three tasks: failure type identification, failure location identification, and GPS quality assessment.

4. Results

The developed multi-task model was trained on the training set with the above-mentioned epochs and batch size. Table 6 represents the performance metrics of the model on the training and testing datasets for the fault type and location identification, respectively.

The trained multi-task ANN model produced 98.89%, 99.12% and 99.24% accuracy for failure type identification, failure location identification and GPS quality assessment. Also, the trained model's precision, recall and AUC values are high, indicating that the model overcomes class imbalance issues.



(a)

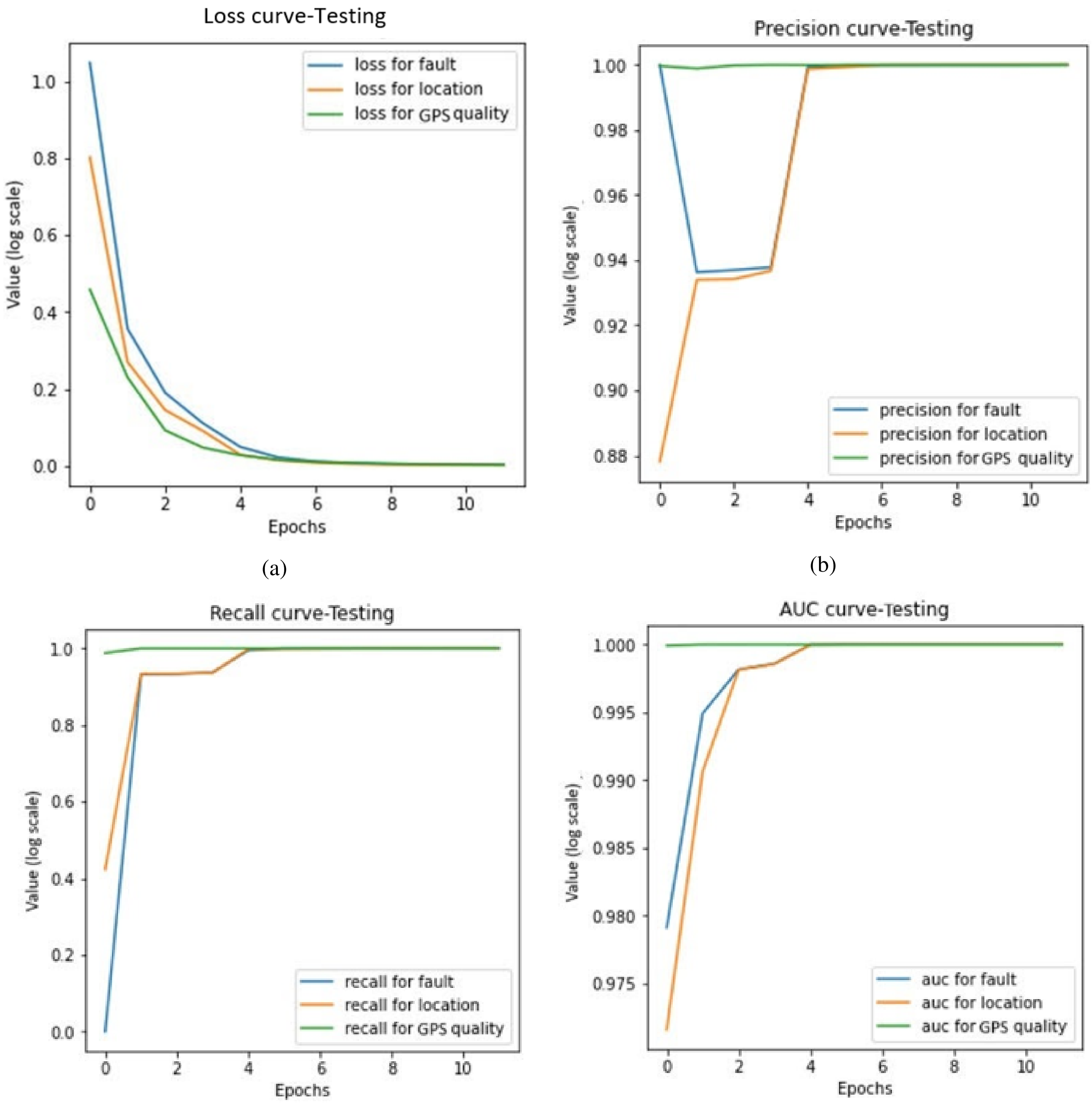
(b)

(c)

(d)

Figure 7. Performance plots of the proposed multi-task model on the training set: (a) loss rate, (b) precision, (c) recall, (d) AUC.

The performance plots representing the values of the performance metrics for each epoch in the training phase are shown in Figure 7. The loss gradually decreases in each task's epoch for fault, location, and GPS quality, as shown in Figure 7(a). In contrast, the recall, AUC, and precision are increasing for each epoch, as shown in Figure 7(b), Figure 7(c), and Figure 7(d), respectively. This shows no fluctuations in the training phase and no signs of varying gradients.



(a)

(b)

(c)

(d)

Figure 8. Performance plots of the proposed multi-task model on the testing set: (a) loss rate, (b) precision, (c) recall, (d) AUC.

The performance plots represent the values of the performance metrics for each epoch in the testing phase. The loss gradually decreases in each task's epoch, as shown in Figure 8(a). In contrast, the recall, AUC, and precision are increasing for each epoch for fault, location, and GPS quality as shown

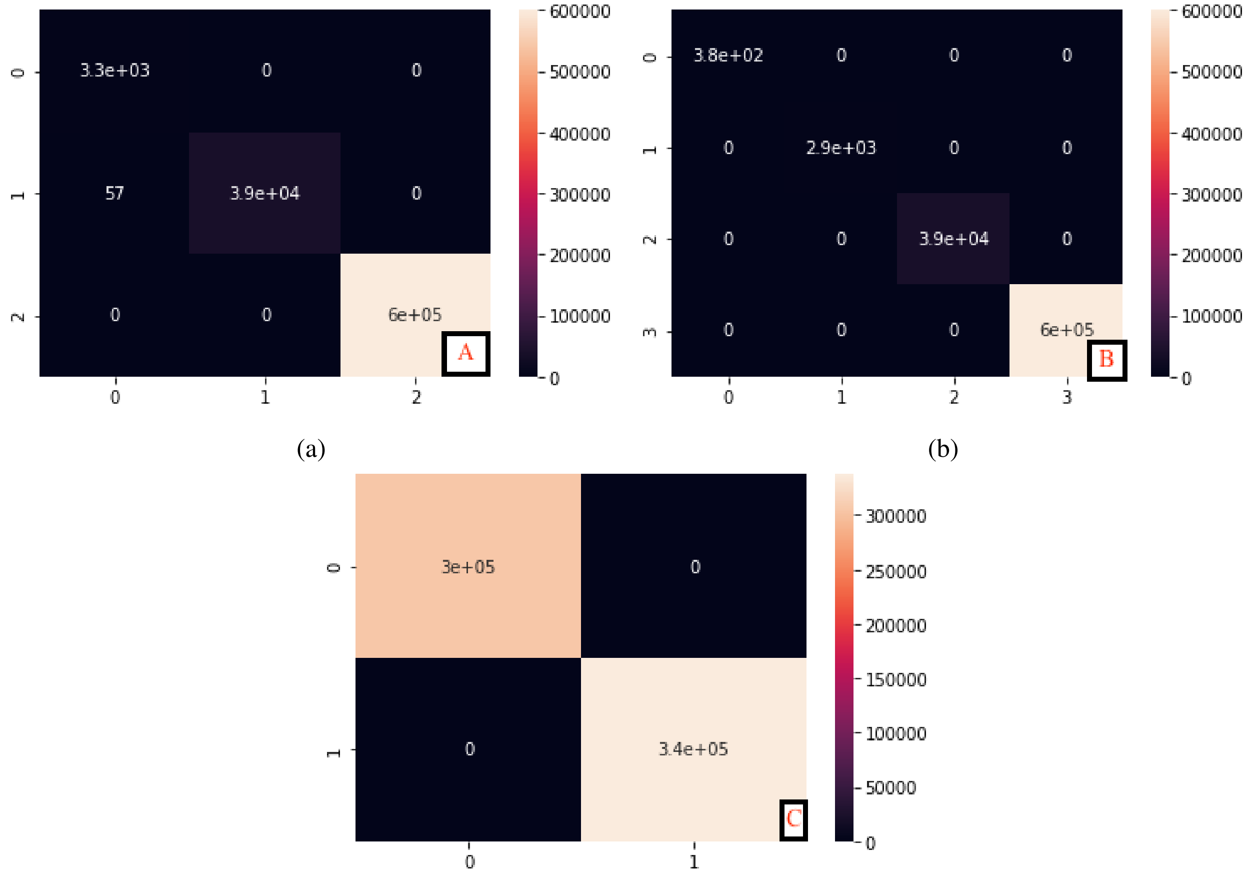


Figure 9. Confusion matrices for the trained multi-task model on the testing sets of all tasks: (a) fault type, (b) fault location, (c) GPS quality.

in Figure 8(b), Figure 8(c), and Figure 8(d). The precision, recall, and AUC values for GPS quality reached the maximum in the initial epochs and remained the same for the rest. Also, fluctuations in the precision values are observed for the tasks of fault type and fault location identification.

The confusion matrices on the testing set represent that the trained model has produced high true negatives and true positives. In contrast, it has produced very few false positives and false negatives. The confusion matrix for the fault type is shown in Figure 9(a), The fault location confusion matrix is shown in Figure 9(b) and the confusion matrix for GPS quality is shown in Figure 9(c).

Table 7. Classification report for the trained multi-task model for fault type identification on the testing set.

Classes	Precision	Recall	F1-Score
0	0.98	1.00	0.99
1	1.00	1.00	1.00
2	1.00	1.00	1.00

The classification report for fault type classification indicates high precision, recall, and F1 scores for all three classes, showing no sign of class imbalance. The classification report for the trained multi-task model for fault type identification on the testing set is shown in Table 7. The classification report for the trained multi-task model for fault location identification on the testing set is shown in Table 8. The classification report for the trained multi-task model for GPS quality assessment on the testing set is shown in Table 9.

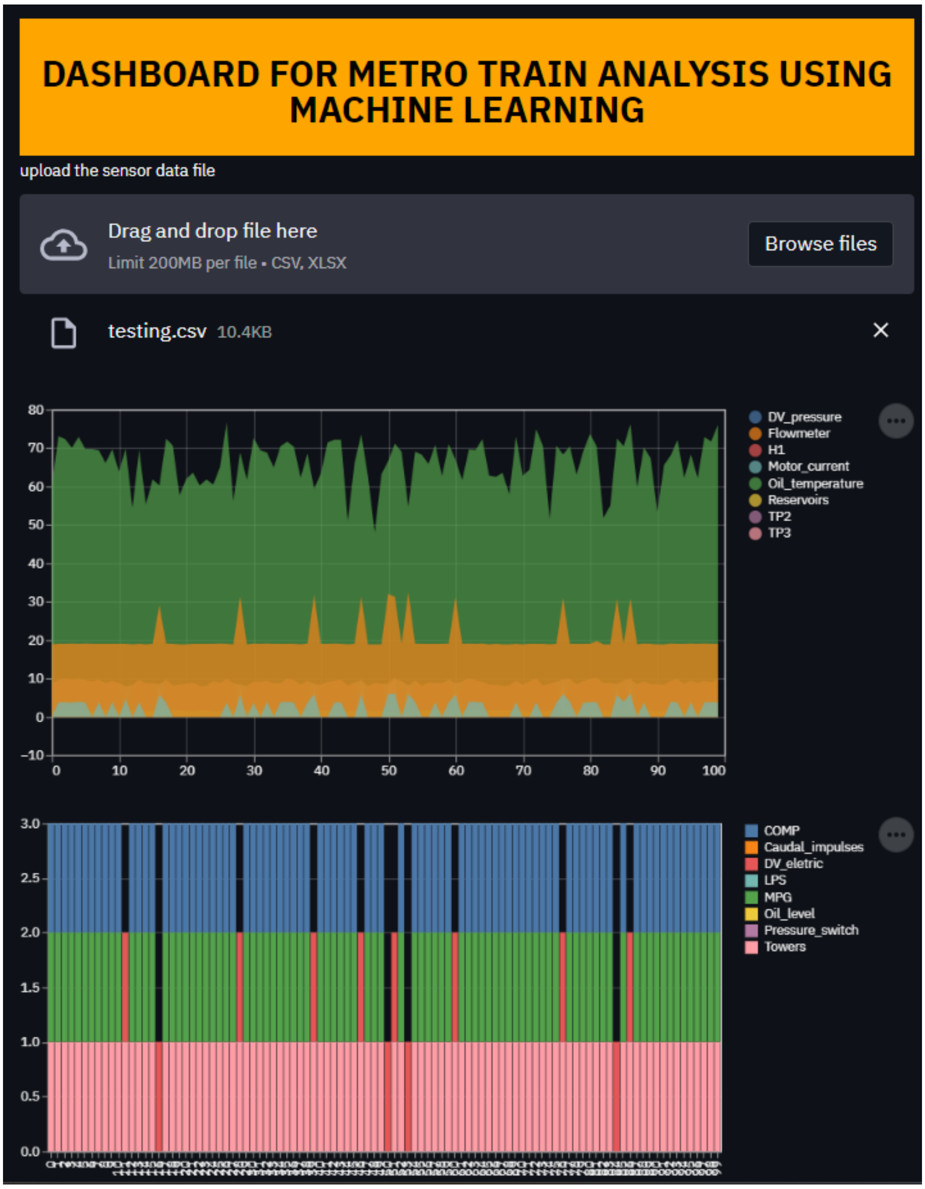
Table 8. Classification report for the trained multi-task model for fault location identification on the testing set.

Classes	Precision	Recall	F1-Score
0	1	1	1
1	1	1	1
2	1	1	1
3	1	1	1

Table 9. Classification report for the trained multi-task model for GPS quality assessment on the testing set.

Classes	Precision	Recall	F1-Score
0	1	1	1
1	1	1	1

A website was developed using Streamlit and hosted online using the Streamlit share, see [21] for the website URL. Figures 10 and 11 represent the snips of the developed website about data visualization and predictive modeling. Figures 12, 13, and 14 represent the LIME explanations for the tasks of failure type identification, failure location identification, and GPS quality assessment, respectively. Figure 10 represents the area chart for the continuous features and the bar chart for the categorical features. From this bar, anomalies in the sensor values can be visually identified. Figure 11 represents the map plot for the GPS data; the dots represent the map's route based on the latitude and longitude values, whereas the dot's intensity depicts the speed. This graph identifies the train's route and the crucial locations at which the train went fast/slow. Also, the multi-task neural network predictions are mentioned on the website for each dataset instance.



DASHBOARD FOR METRO TRAIN ANALYSIS USING MACHINE LEARNING

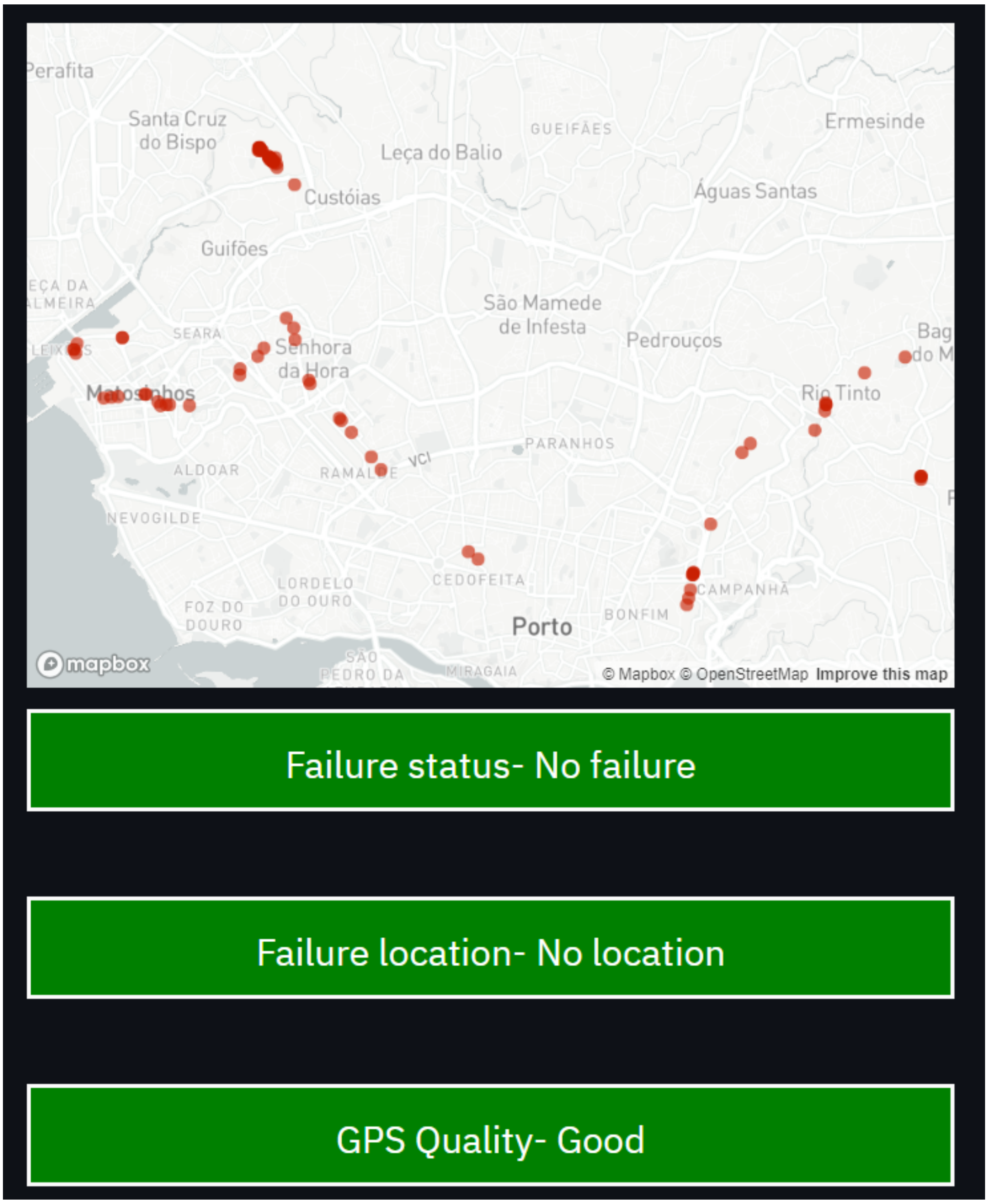
upload the sensor data file

Drag and drop file here

Limit 200MB per file • CSV, XLSX

testing.csv 10.4KB

Figure 10. The data visualization part of the developed website shows the plot for continuous and categorical features of the testing data.

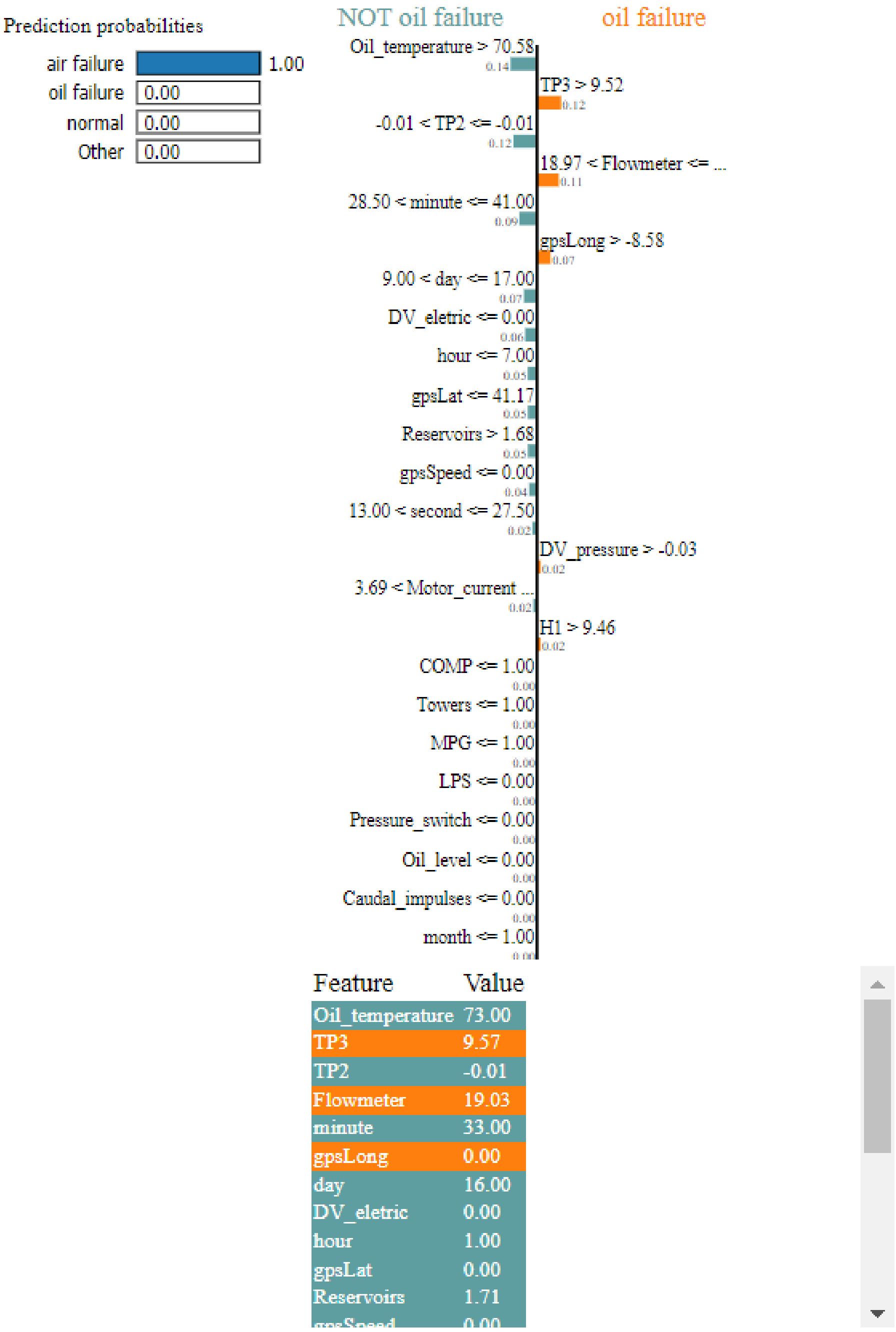


Failure status- No failure

Failure location- No location

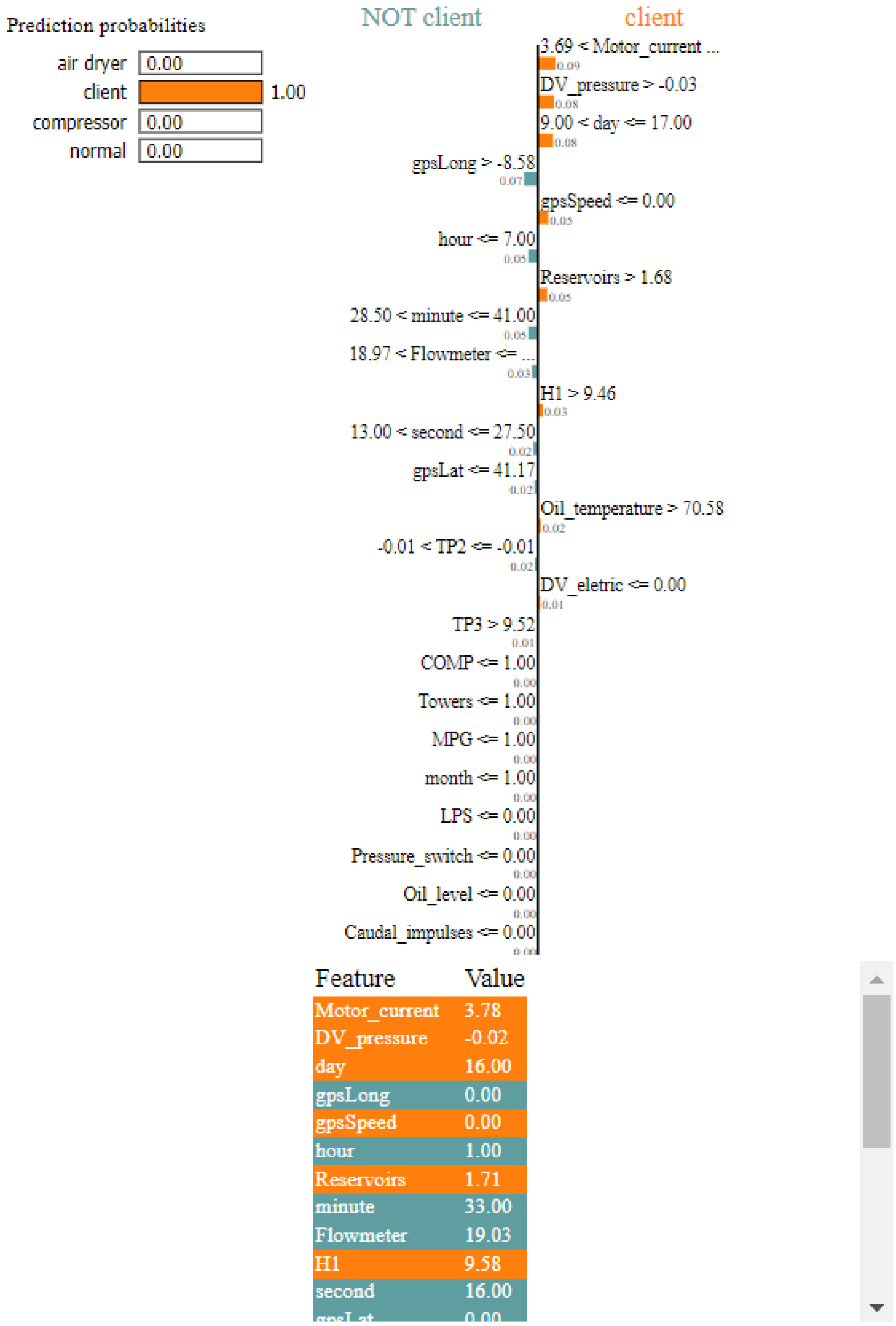
GPS Quality- Good

Figure 11. Map representing the speed and route of the train along with the predictions made by the multi-task model.



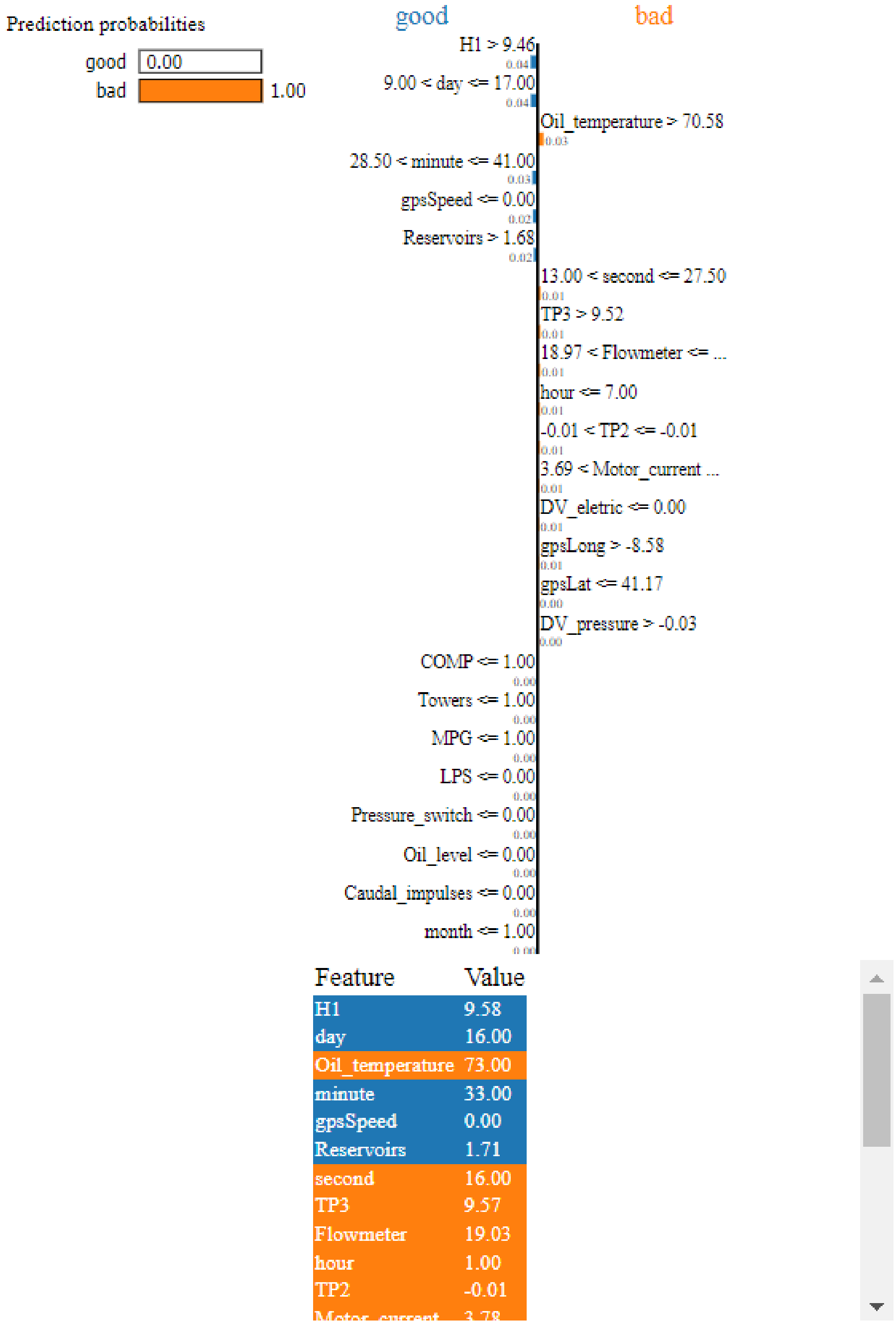
Feature	Value
Oil_temperature	73.00
TP3	9.57
TP2	-0.01
Flowmeter	19.03
minute	33.00
gpsLong	0.00
day	16.00
DV_eletric	0.00
hour	1.00
gpsLat	0.00
Reservoirs	1.71
gpsSpeed	0.00

Figure 12. LIME explanations for the model prediction on random samples for failure type identification.



Feature	Value
Motor_current	3.78
DV_pressure	-0.02
day	16.00
gpsLong	0.00
gpsSpeed	0.00
hour	1.00
Reservoirs	1.71
minute	33.00
Flowmeter	19.03
H1	9.58
second	16.00
gpsLat	0.00

Figure 13. LIME explanations for the model prediction on random samples for failure location identification.



Feature	Value
H1	9.58
day	16.00
Oil_temperature	73.00
minute	33.00
gpsSpeed	0.00
Reservoirs	1.71
second	16.00
TP3	9.57
Flowmeter	19.03
hour	1.00
TP2	-0.01
Motor_current	3.78

Figure 14. LIME explanations for the model prediction on random samples for the task of GPS quality assessment.

Table 10. Comparative analysis of the results of the proposed work to that of the existing works.

Author	Algorithm	Result
Rajashekarappa et al. [13]	RUS Boosted Classifier	98.73% accuracy
Najjar et al. [15]	Random Forest Classifier	97% accuracy
Davari et al. [16]	Autoencoder	44% improvement in precision compared to baseline
Proposed work	Multi-task Artificial Neural Network	99% average accuracy

Table 10 compares the performance of the proposed multi-task neural network to that of the existing works related to the application of predictive maintenance in train fault analysis using sensor data.

5. Discussion

Based on our knowledge, only two works [15, 16] use artificial intelligence for the predictive maintenance of urban metro trains. Najjar et al. [15] worked on predicting air failure of the air production unit (APU) in metro trains. The dataset used for this task was MetroPT, a 6-month analysis of metro trains in Portugal comprising analog, digital, and GPS sensors. The GPS information was excluded from the dataset, and the timestamp was encoded using the label encoding technique. A random forest classifier algorithm was used for the multi-class classification of air failure prediction. The data was undersampled and then split into training and testing sets. A feature importance visualization technique was employed to identify the root cause of the air failure. The proposed work produced testing accuracies of 84% and 87% on the binary and multi-class classification tasks and F1 scores in the ranges of 0.83–0.5 and 0.73–0.97 for the binary and multi-class classification tasks. The proposed model produced better results than the work and has also included oil failures in addition to air failures. Davari et al. [16] developed a deep learning neural network for anomaly detection in metro trains. The algorithms used for this task were the sparse autoencoder and variational autoencoder. This work is an unsupervised learning approach for anomaly detection of air failures in trains. The sparse autoencoder trained on the digital data produced 42% more than those trained on analog data. Also, the variational autoencoder performed better than the sparse autoencoder by 37%. The proposed work considered both analog, digital, and GPS sensors worked on both air and oil failures, and produced state-of-the-art results.

The multi-task model has produced excellent results on the training and testing datasets. The performance plots prove that the model has been trained perfectly and shows no signs of overfitting or underfitting. Also, the confusion matrices and classification report suggest that the trained model is generalized and does not exhibit any signs of class imbalance.

The proposed work has some advantages in comparison to the existing works. The proposed work addresses all issues faced in the metro trains (air and oil failures). Another advantage is the excellent results obtained by the trained models. The third advantage is agility; the proposed multi-task model has taken less time to train and predict batch data. Finally, the explainable AI technique, namely LIME, has been implemented to provide interpretations to the outputs given by the multi-task model. This provides belief in the model prediction and can also be useful for engineers to deeply analyze the issue.

Figure 12 represents the LIME explanation for a local instance related to the task of fault type prediction. As observed from the plot, the predicted instance is air failure with 100% confidence, and the features positively and negatively contribute to the prediction.

Figure 13 represents the LIME explanation for a local instance related to the task of fault location prediction. As observed from the plot, the predicted instance is a client with 100% confidence, and the features positively and negatively contribute to the prediction. Motor current, DV pressure, and the day positively contributed to the prediction, followed by GPS speed and reservoir. The GPS longitude, minute, and flowmeter have negatively contributed to the prediction.

Figure 14 represents the LIME explanation for a local instance related to the task of GPS quality prediction. As observed from the plot, the predicted instance is air failure with 100% confidence, and the features positively and negatively contribute to the prediction. Oil temperature has majorly contributed positively, whereas H1, day, and minute have majorly contributed negatively. For all plots, the range or condition of the input features is given, which might be of great use for the fault analysis.

One reason for achieving good results is the split of the output labels into fault type and location. This reduces the number of interdependent classes in each stage, which might have improved the performance of the algorithms. Also, we observed a performance rise of 6% when the features were standardized. Considering the deep learning aspects, we developed a multi-task model which splits the classes into individual tasks, allowing for more attention, resulting in better results and quicker periods.

However, the study has some limitations. First, the dataset was undersampled to 30,00,000 data points, roughly 20% of the entire dataset. The second one was the expansion of the target vectors into new columns, which introduced more computations and the need to train more models. While this approach was considered to improve the holistic performance of the models, this resulted in the creation of multiple datasets and, ultimately, multiple ML models for training, leading to computational costs. The third one was the limited selection of machine learning algorithms. Many good machine learning algorithms like support vector machine and ensemble learning techniques were not implemented due to the computational constraints and long training durations (the SVM algorithm did not train even after 30 minutes!).

6. Conclusions

Hence in this work, a multi-task model was developed for the identification of failures simultaneously. The proposed method has produced 98.89%, 99.12%, and 99.24% accuracies in the testing set for failure type, failure location, and GPS quality predictions, respectively, exceeding the state-of-the-art methods. The model produced 99.56%, 99.67% ,and 99.84% precision in the testing set for failure type, failure location, and GPS quality predictions, respectively. The high accuracy and precision values indicate the good performance of the model and no signs of class imbalance. The deep learning model took 43 seconds for training and 1 second for inferencing for test data, showing fast predictions, needed for predictive maintenance applications. Moreover, a real-time interactive dashboard was developed, performing dynamic data visualization and predictions. Finally, using the LIME explainable AI technique provides explanations for the predictions, adding belief and better analysis for engineers. The developed system would be advantageous for engineers to perform fault analysis and predictive maintenance effectively. Future work will use a database to store the streaming data and deploy the system in real time. Also, we will develop deep learning algorithms on the entire dataset and employ online learning strategies to update the learned model in real time.

Author Contributions

Pratik Vinayak Jadhav: Data Curation and Analysis, Research Design and Methodology, writing draft. Sairam V. A: Data Curation and Analysis, Research Design and Methodology, writing draft. Siddharth Sonkavade: Data Curation and Analysis, Research Design and Methodology, writing draft. Shivali Amit Wagle : Conceptualization, Supervision, Project Development, Writing, Review, and Editing. Preksha Pareek: Conceptualization, Supervision, Project Development, Writing, and Review. Ketan Kotecha: Writing, Review, and Editing, Funding and Resources. Tanupriya Choudhury: Data Analysis, Writing, Review, and Editing.

Conflict of interest

The authors declare no conflict of interest.

References

1. Muchiri P, Pintelon L. (2008) Performance measurement using overall equipment effectiveness (OEE): literature review and practical application discussion. *Int J Prod Res* 46: 3517–3535. https://doi.org/10.1080/00207540701290454
2. Pashami S, Nowaczyk S, Fan Y, Jakubowski J, Paiva N, Davari N, Bobek S, Jamshidi S, Sarmadi H, Alabdallah A, et al. (2023) Explainable Predictive Maintenance. *arXiv preprint* arXiv:2306.05120.
3. Mallia J, Francalanza E, Xuereb PA, Refalo P. (2023) Intelligent Approaches for Anomaly Detection in Compressed Air Systems: A Systematic Review. *Machines* 11: 750. https://doi.org/10.3390/machines11070750
4. Ding J, Zuo J. (2023) Analysis of leakage fault characteristics of air control valves in train braking system. In: *Sixth International Conference on Traffic Engineering and Transportation System (ICTETS 2022)*, SPIE, 12591: 71–78.
5. Tian W, Wang W, Lu Z, Wang Z, Hua W, Cheng M. (2022) Collaborative control for half-centralized open-end winding permanent-magnet linear motor drive systems. *IEEE Trans Power Electron* 37: 10399–10411. https://doi.org/10.1109/TPEL.2022.3143292
6. Pejić Bach M, Topalović A, Krstić Ž, Ivec A. (2023) Predictive Maintenance in Industry 4.0 for the SMEs: A Decision Support System Case Study Using Open-Source Software. *Designs* 7: 98. https://doi.org/10.3390/designs7040098
7. Çınar ZM, Nuhu AN, Zeeshan Q, Korhan O, Asmael M, Safaei B. (2020) Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability* 12: 8211. https://doi.org/10.3390/su12198211
8. Xu G, Liu M, Wang J, Ma Y, Wang J, Li F, Shen W. (2019) Data-Driven Fault Diagnostics and Prognostics for Predictive Maintenance: A Brief Overview. In: *15th IEEE International Conference on Automation Science and Engineering (CASE)*, 103–108. https://doi.org/10.1109/COASE.2019.8843068
9. Xu H, Sun Z, Cao Y, Bilal H. (2023) A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Comput* 27: 14469–14481. https://doi.org/10.1007/s00500-023-09037-4

10. Kotsiopoulos T, Sarigiannidis P, Ioannidis D, Tzovaras D. (2020) Machine Learning and Deep Learning in Smart Manufacturing: The Smart Grid Paradigm. *Comput Sci Rev* 40: 100341. https://doi.org/10.1016/j.cosrev.2020.100341

11. Sun X, Ling KV, Sin KK, Liu Y. (2021) Air Leakage Detection of Pneumatic Train Door Subsystems Using Open Set Recognition. *IEEE Trans Instrum Meas* 1–1. https://doi.org/10.1109/TIM.2021.3096267

12. Lee WJ. (2020) Anomaly Detection and Severity Prediction of Air Leakage in Train Braking Pipes. *Int J Progn Health Manag* Available from: https://api.semanticscholar.org/CorpusID:204779350

13. Rajashekarappa M, Lene J, Turanoğlu Bekar E, Skoogh A, Karlsson A. (2021) A Data-Driven Approach to Air Leakage Detection in Pneumatic Systems. In: *2021 Prognostics and Health Management Conference (PHM)*, 1–7. https://doi.org/10.1109/PHM-Nanjing52125.2021.9612973

14. Sulaiman N, Abdullah MP, Abdullah H, Zainudin M, Yusop A. (2020) Fault detection for air conditioning system using machine learning. *IAES Int J Artif Intell* 9: 109–116. https://doi.org/10.11591/ijai.v9.i1.pp109-116

15. Najjar A, Ashqar H, Hasasneh A. (2023) Predictive Maintenance of Urban Metro Vehicles: Classification of Air Production Unit Failures Using Machine Learning. *In press*.

16. Davari N, Veloso B, Ribeiro R, Pereira P, Gama J. (2021) Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In: *7th IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 1–10. https://doi.org/10.1109/DSAA53316.2021.9564181

17. Veloso B, Ribeiro R, Gama J, Pereira P. (2022) The MetroPT dataset for predictive maintenance. *Sci Data* 9: Article 1877. https://doi.org/10.1038/s41597-022-01877-3

18. O, Al Mamlook RE, Shehadeh A, Munir T. (2024) Empirical exploration of predictive maintenance in concrete manufacturing: Harnessing machine learning for enhanced equipment reliability in construction project management. *Comput Ind Eng* 110046. https://doi.org/10.1016/j.cie.2024.110046

19. Alshboul O, Almasabha G, Shehadeh A, Al-Shboul K. (2024) A comparative study of LightGBM, XGBoost, and GEP models in shear strength management of SFRC-SBWS. In: *Structures* 61: 106009. https://doi.org/10.1016/j.istruc.2023.106009

20. Hung YH, Lee CY. (2024) BMB-LIME: LIME with modeling local nonlinearity and uncertainty in explainability. *Knowl Based Syst* 294: 111732. https://doi.org/10.1016/j.knosys.2023.111732

21. Jadhav PV, Sairam VA, Sonkavade S. (2023) Predictive Maintenance Dashboard. Available from: https://dashboard-for-predictive-maintenance.streamlit.app