Data Mining Project: MetroPT-3 Predictive Maintenance

Malak Gaballa
Masa Tantawy
Moustafa El Mahdy

Table of Contents

Introduction **2**

Labelling 5

Exploratory Data Analysis (EDA) 6

Methodology **9**

Techniques Description 10

Evaluation 13

Results **14**

Approach 1: Full Data 14

Approach 2: PCA (2 components) on full data 15

Approach 3: Random Sample (Half the data, same ratio) 15

Approach 4: Balanced Data 16

Conclusion **18**

Appendix **19**

Introduction

Predicting Air Production Unit (APU) failures in metro train operations is a critical challenge with profound implications for maintenance efficiency and operational reliability. Traditionally, preventive and reactive maintenance strategies have been employed, often relying on predefined schedules or post-failure responses. However, the dynamic nature of APU failures necessitates a more proactive and anticipatory approach.

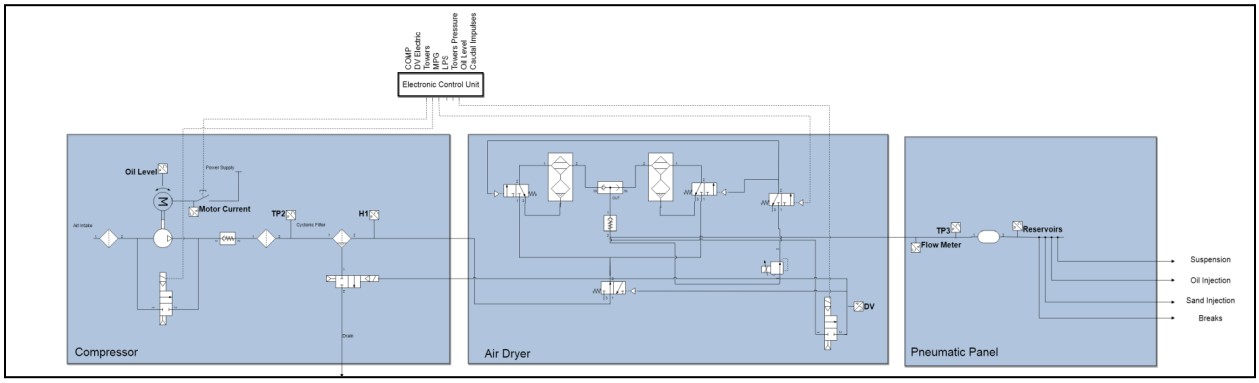
The MetroPT-3 dataset is a valuable repository encapsulating the heartbeat of metro train operations. Derived from real-world readings within a compressor's Air Production Unit (APU), this dataset provides a comprehensive insight into the operational dynamics of metro systems. Collected parameters, including pressure, temperature, motor current, and air intake valves, offer a granular perspective on the intricate interplay of factors influencing APU functionality. This dataset, derived from real-world operations, not only showcases the difficulties involved in predictive maintenance within the metro train sector but also demonstrates the genuine nature of the data, establishing it as an indispensable asset for the creation and evaluation of predictive models. Its usefulness goes beyond predicting failures alone, providing a foundation for anomaly detection and other essential tasks that contribute to improving the dependability and effectiveness of metro train systems.

The dataset consists of 1,516,948 instances, representing measurements from 15 sensors of a metro compressor's APU collected at 1Hz from February to August 2020. The dataset can be characterized as Tabular, Multivariate, and Time-Series. A description of the dataset, including feature names and types is shown in the table below.

Feature	Feature Type	Sensor Type	Unit	Description
timestamp	DateTime	-	-	Time that sensors collect data from February to August 2020.
TP2 (Throttle Position 2)	Continuous	Analogue	Bars	Measure of the pressure on the compressor. It indicates the pressure level at which the compressor is operating.
TP3 (Throttle Position 2)	Continuous	Analogue	Bars	Measure of the pressure generated at the pneumatic panel.
H1 (HydroLogger)	Continuous	Analogue	Bars	Measure of the pressure generated due to pressure drop when the discharge of the cyclonic separator filter occurs. It indicates the pressure change caused by the filter.
DV_pressure	Continuous	Analogue	Bars	Measure of the pressure drop generated when the towers discharge air dryers; a zero reading indicates that the compressor is operating under load.
Reservoirs	Continuous	Analogue	Bars	Measure of the downstream pressure of the reservoirs. It should be close to the pneumatic panel pressure (TP3) and indicate the pressure level in the reservoirs.
Motor_current	Continuous	Analogue	Amps	Measure of the current of one phase of the three-phase motor; it presents values close to ● 0A - when it turns off, ● 4A - when working offloaded ● 7A - when working under load ● 9A - when it starts working.
Oil_temperature	Continuous	Analogue	ºC	Measure of the oil temperature on the compressor.
COMP	Binary	Digital	-	The electrical signal of the air intake valve on the compressor. ● 1 = active when there is no air intake, indicating that the compressor is either turned off

				or operating in an offloaded state.
DV_electric	Binary	Digital	-	The electrical signal that controls the compressor outlet valve. ● 1 = active when the compressor is functioning under load ● 0= inactive when the compressor is either off or operating in an offloaded state.
TOWERS	Binary	Digital	-	The electrical signal that defines the tower responsible for drying the air and the tower responsible for draining the humidity removed from the air. ● 0 = tower one in operation ● 1 = tower two in operation
MPG	Binary	Digital	-	The electrical signal responsible for starting the compressor under load by activating the intake valve when the pressure in the air production unit (APU) falls below 8.2 bar; it activates the COMP sensor, which assumes the same behavior as the MPG sensor.
LPS (Low Pressure Sensor)	Binary	Digital	-	The electrical signal that detects and activates when the pressure drops below 7 bars. It is used as a pressure safety mechanism.
Pressure_switch	Binary	Digital	-	The electrical signal that detects the discharge in the air-drying towers.
Oil_level	Binary	Digital	-	The electrical signal that detects the oil level on the compressor. ● 1 = oil is below the expected values.
Caudal_impulses	Binary	Digital	-	An electrical signal that counts the pulse outputs generated by the absolute amount of air flowing from the APU to the reservoirs.

To be able to visualize the sensors that are used, a visual aid for the Air Production Unit (APU) is provided below.



This project aims to identify instances of APU failures thus requiring maintenance. To effectively predict failures and spot issues in metro train systems, it's important to use data mining techniques that involve using advanced algorithms to find hidden patterns and irregularities in large datasets, including historical maintenance records, sensor data, and operational logs. To achieve this, multiple data mining techniques in addition to machine learning classifiers were used to be able to detect instances with APU failures. These will be explained in detail in the next section.

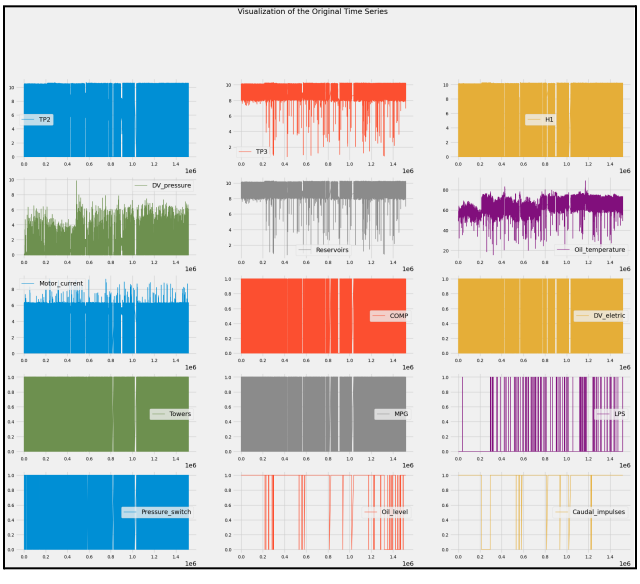
Labelling

It is important to note that the initial data was unlabeled hence there was no label/ attribute to describe if the instance had APU failure and thus required maintenance or not. After extensive research, we have come across the table below, which shows that the air leak instances had happened only 4 times. Accordingly, all the data points that fall within the stated timestamps have an air leak, and so have a value = 1 in the “Airleak” column. Now that the timestamp was no longer necessary, this attribute was dropped from the dataset in addition to duplicates. Instances with air leaks constituted 29,865 out of 1,459,475 which is approximately 2.05% of the data.

Nr.	Start Time	End Time	Failure	Severity	Report
#1	4/18/2020 0:00	4/18/2020 23:59	**Air leak**	High stress	
#1	5/29/2020 23:30	5/30/2020 6:00	**Air Leak**	High stress	Maintenance on 30Apr at 12:00
#3	6/5/2020 10:00	6/7/2020 14:30	**Air Leak**	High stress	Maintenance on 8Jun at 16:00
#4	7/15/2020 14:30	7/15/2020 19:00	**Air Leak**	High stress	Maintenance on 16Jul at 00:00

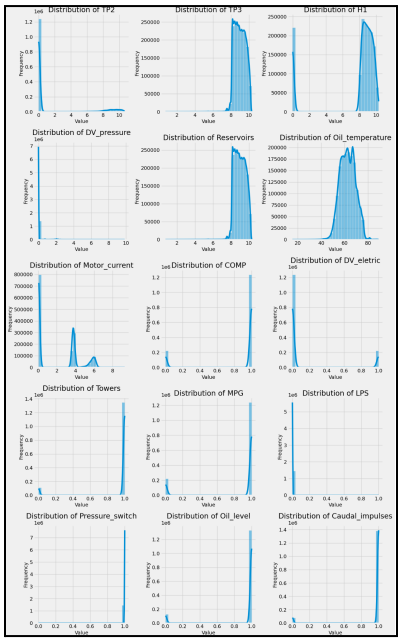
Exploratory Data Analysis (EDA)

Before applying data mining and machine learning techniques to identify air leaks, a comprehensive exploratory data analysis (EDA) was conducted to ensure a thorough understanding of the dataset. Since the data did not initially include qualitative features, preprocessing was unnecessary in that regard. However, the timestamp was removed because it wasn't significant in detecting air leaks and 57,473 duplicate instances were found which were removed as well, ending up with 1,459,475 instances and 16 features. A thorough check for missing values was performed, and fortunately, none were found.

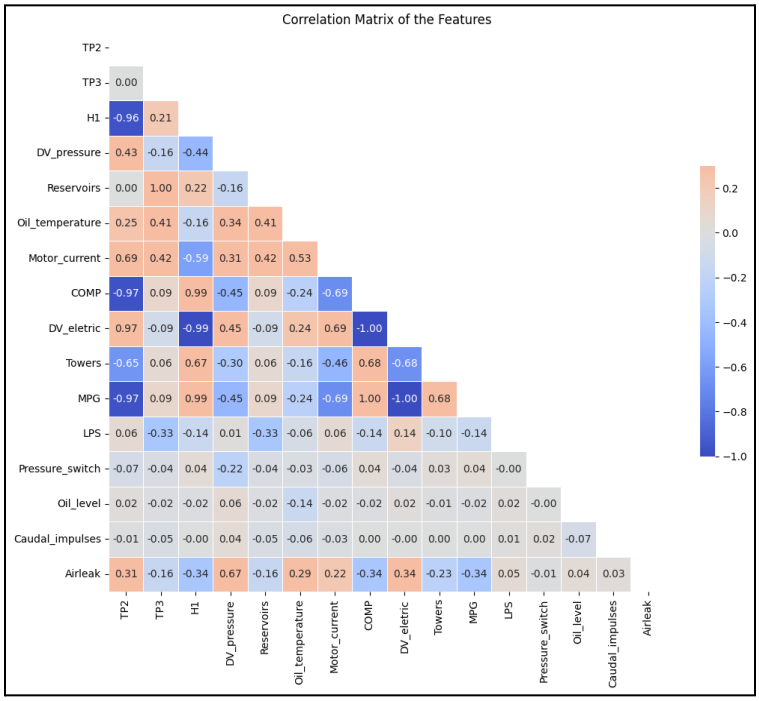


Time series plots were then constructed for each feature, covering the period from February 2020 to August 2020, aiming to detect any discernible trends or patterns. The majority of features had fluctuating trends to the point where the whole plots are colored as shown in the accompanying graph. Additionally, distribution

plots were generated to visualize the densities of each feature. The graphs are explainable since 8 out of 15 features are binary which makes the distribution peaking at the values 0 and 1 as shown in the graph. Some features are skewed such as TP3 , H1, and Reservoirs while the feature Oil_temperature generated a normally distributed bell curve. From the distributions, it is evident that the features TP3 and Reservoirs have identical densities, which is explained by the data description provided.



A correlation matrix was also constructed, prompted by observations in the time series plots that revealed certain features exhibited consistent trends within the specified time frame. Notably, features like TP3 and Reservoirs, as well as COMP and MPG, displayed perfect positive correlations of 1 and 0.98, respectively. Although tempting to remove one from each pair, it was reasoned that these features are integral to the system, indicating potential air leaks.



Looking at the correlations between the features and the label “Airleak”, it was evident that DV Electric exhibited a high correlation coefficient of 0.67 with the label. This implies that when the compressor operates under load (equal to 1), the likelihood of an air leak increases.

With EDA completed, the next steps involve implementing the approaches outlined in the subsequent sections for effective identification and mitigation of air leaks.

Methodology

As previously explained, our goal is to successfully identify instances with APU failure thus having an air leak. To achieve this, data mining approaches were applied to 4 different variants of the dataset, each constituting an approach, which are:

1. Applications on the full dataset
2. Applications on 2 principle components on the full dataset
3. Applications on a random sample of the dataset preserving the ratio of anomalies
4. Applications on a balanced dataset

These different approaches stemmed out of the need to reduce the massive dataset size to allow data mining and machine learning algorithms to operate without the need for high GPUs. In the first approach, the data consisted of 1,459,475 observations and 15 attributes in addition to the label. In the second approach, we attempted a dimensionality reduction technique to make the dataset smaller; the data consisted of 1,459,475 observations, and 2 PCA components, which captured the most variance in the dataset, in addition to the label. Due to the need to further reduce the dataset's size, in the third approach, only half the original dataset was used while ensuring that the ratio of observations with air leak remained the same; the data consisted of 729,738 observations with 2.07 % with air leak, and 15 attributes in addition to the label. Finally, in the fourth approach, a data-imbalance handling technique was adopted; all observations with air leaks were used in addition to a random sample of those without air leaks of the same size. Hence, the data was equally balanced, and it consisted of 59,730 observations and 15 attributes in addition to the label.

Initially, observations with air leaks were treated as anomalies due to the extremely small number in approaches 1, 2, and 3 therefore 3 anomaly detection approaches were applied:

clustering-based , distance-based , and density-based approaches. Each approach had multiple techniques; specifically, for the clustering-based approach, the attempted techniques were K-Means clustering using the Euclidean distance, Hierarchical clustering using different linkage measures, and Spectral Clustering. For the distance-based approach, the attempted techniques were distance to the fifth-nearest neighbor using the Euclidean distance, Mahalanobis distance, and BACON. For the density-based approach, DBSCAN: Density-Based Spatial Clustering of Applications with Noise was implemented with various parameters (epsilon and min_samples).

In the fourth approach, instances with air leaks are no longer considered anomalies as they constitute half of the data. To identify them, 2 approaches were adopted which are clustering and machine learning classifiers. The clustering approach involved K-Means clustering using the Euclidean distance, Hierarchical clustering using different linkage measures and Spectral Clustering, and machine learning classifiers. The machine learning classifiers used were Random Forests, K-NN, Logistic Regression, SVM, Least Squares, and Perceptron. It is worth noting that using machine learning classifiers in the first 3 approaches (on imbalanced datasets) would have had very weak performance since the models would be biased as a result of the extreme data imbalance.

Techniques Description

Clustering is an unsupervised learning technique where we aim to group similar data points together based on their inherent patterns or similarities. It has several techniques as *K-Means* which is an algorithm that separates a dataset into k clusters by assigning each data point to the cluster whose centroid is the closest, iterating until convergence. Additionally, *Hierarchical Clustering*, from its name, creates a hierarchy of clusters by merging or splitting existing clusters based on their similarity. It can be agglomerative (bottom-up) or divisive

(top-down), and it relies on distance measures and linkage techniques such as single linkage, complete linkage, average linkage, ward and others to separate clusters based on intra-cluster and inter-cluster distances. Lastly, *Spectral Clustering* relies on transforming the data into a lower-dimensional space using the eigenvalues and eigenvectors of a similarity matrix, then applies a traditional clustering algorithm, such as K-Means, to partition the transformed data into clusters.

The distance-based approach for anomaly detection identifies anomalies in a dataset by measuring the distances between data points. Its first technique that we used was *5-NN* which assigns a score to each data point based on its distance from its fifth-nearest neighbour using euclidean distance. The threshold used to classify points as outliers is set based on the distance value that yields a percentage of outliers equal to or more than the known percentage of outliers, which is approximately 2% in this case. As for the *Mahalanobis Distance*, it measures the distance of each data point from the centroid of the dataset, taking into account the correlation between features in the data; points with higher Mahalanobis distances, which are greater than a value set using the chi square distribution, are considered anomalies/outliers since they deviate from the expected distribution. Lastly, *BACON*, Blocked, Adaptive, Computationally-Efficient Outlier Nominator, is an algorithm for detecting anomalies in multivariate space by iteratively identifying and removing data points that contribute the most to the covariance matrix, taking into consideration both distance and statistical measures. This is done to help identify outliers and anomalies that may distort the data.

DBSCAN, Density-Based Spatial Clustering of Applications with Noise, is a clustering method that separates clusters of high density from clusters of low density. It distinguishes between main points, which have a sufficient number of neighbours (exceeding the set threshold

of minimum samples) within a specified radius (known as epsilon), and noise points that do not belong to any cluster. To clarify, high-density clusters are areas of the dataset with multiple data points close to each other, and points in low-density areas are considered outliers since they are far from any cluster. The parameters epsilon and min_samples were varied until the percentage of anomalies yielded was equal to or greater than the known percentage, which is 2% in this case, as done with the fifth-nearest neighbour technique.

For the machine learning classifiers, multiple models were used. *Random Forest* is an ensemble learning method that constructs several decision trees during training and combines their output to reach a single result. *K-NN* is a supervised learning algorithm that classifies a data point based on the majority class of its k nearest neighbours in the feature space. This is done by calculating the distance between each point and its k nearest neighbours then classifying it based on the majority class of these points, making it simple and effective for classification. *Logistic Regression* is a statistical algorithm used for classification by predicting the probability of an outcome, event, or observation using the log-odds for the event which is calculated as a linear combination of one or more independent variables. *SVM*, Support Vector Machines, maps data to a high-dimensional feature space/hyperplane so that data points can be well separated thus categorised, which is useful when the data are not otherwise linearly separable. *Least Squares* is a method that minimises the sum of squared differences between the observed and predicted values. It is used to find the coefficients of a linear model that best fits the given data. Finally, the *Perceptron* is a simple form of a neural network used for binary classification tasks. It takes input features which can be non-linearly transformed to afford separability, applies weights using an activation function, and produces an output that is compared to the true label. The model is trained to adjust weights based on misclassifications.

Evaluation

To evaluate all of the aforementioned techniques in each approach, a set of metrics were used. Given that the initial data has more observations with no air leak than those with air leak, the accuracy would have always been high. Hence, more reliable metrics were necessary; the ones used were precision, recall, and F-score. A description of each metric is shown below. For the machine learning classifiers, these metrics were based on cross-validation using a 5-fold.

- **Recall:** This is the ratio, known as true positive rate (TPR), of correctly predicted positive points out of all actual positive points in the dataset. It ranges from 0 to 1; the higher the better.

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

- **Precision:** This is the ratio, known as false positive rate (FPR), of correctly predicted positive points out of all predicted positive points in the dataset. It ranges from 0 to 1; the higher the better.

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

- **F1-Score:** This is a weighted average of both the recall and precision and is useful when one of these two metrics is high and the other is low. In such scenarios, the F1-score takes both into account while punishing extreme values since it uses the harmonic mean and not the arithmetic mean. It ranges from 0 to 1; the higher the better.

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

Results

In this section, the results of the different techniques explained earlier are illustrated for the four approaches. It is worth highlighting that multiple techniques failed due to limitations such as GPU availability and the large dataset size, which led to trying these techniques in the next approaches.

Approach 1: Full Data

1. **Clustering**

	Precision	**Recall**	**F1-score**
K-means	0.05	1.00	0.09
Hierarchical Clustering	Crashed		
Spectral Decomposition	Crashed		

2. **Distance-based Approach for Anomaly Detection**

	Precision	**Recall**	**F1-score**
Fifth Nearest Neighbour	0.09	0.18	0.12
Mahalanobis Distance	0.21	1.0	0.35
BACON	0.07	1.00	0.14

3. **Density-based Approach for Anomaly Detection**

DBSCAN (Density Based Spatial Clustering of Applications with Noise)

	Precision	**Recall**	**F1-score**
eps= 0.2, min_samples= 4	Crashed		
eps= 0.2, min_samples= 3	Crashed		

Approach 2: PCA (2 components) on full data

1. **Clustering**

	Precision	Recall	F1-score
K-means	0.13	1.00	0.24
Hierarchical Clustering	Crashed		
Spectral Decomposition	Crashed		

2. **Distance-based Approach for Anomaly Detection**

	Precision	Recall	F1-score
Fifth Nearest Neighbour	0.13	0.17	0.14
Mahalanobis Distance	0.20	1.00	0.33
BACON	0.06	1.00	0.11

3. **Density-based Approach for Anomaly Detection**

DBSCAN (Density Based Spatial Clustering of Applications with Noise)

	Precision	Recall	F1-score
eps= 0.2, min_samples= 4	Crashed		

Approach 3: Random Sample (Half the data, same ratio)

1. **Clustering**

	Precision	Recall	F1-score
K-means	0.05	1.00	0.09
Hierarchical Clustering	Crashed		
Spectral Decomposition	Crashed		

2. **Distance-based Approach for Anomaly Detection**

	Precision	Recall	F1-score
Fifth Nearest Neighbour	Crashed		
Mahalanobis Distance	0.22	1.00	0.35
BACON	0.07	1.00	0.13

3. **Density-based Approach for Anomaly Detection**

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

	Precision	Recall	F1-score
eps= 0.2, min_samples= 4	Crashed		
eps= 0.2, min_samples= 3	0.04	0.05	0.05

Approach 4: Balanced Data

1. **Clustering**

	Precision	Recall	F1-score
K-means	0.92	1.00	0.96
Hierarchical Clustering	Crashed		
Spectral Decomposition	Crashed		

2. **Machine Learning Classifiers**

Results are based on cross-validation on 5-fold

	Precision	Recall	F1-score
Random Forest	1.00	0.93	0.96
K-NN (3-NN)	0.99	0.93	0.96
Logistic Regression	0.98	0.94	0.96
SVM	0.98	1.00	0.99

Least Squares	0.98	0.94	0.96
Perceptron	0.98	0.98	0.98

As visible from the figures, the machine learning classifiers performed the best among all the applied techniques and approaches. The precision, recall, and F1-score are extremely high, almost 1 all the time, based on cross-validation using a 5-fold. This indicates that these classifiers do not overfit, and they can accurately and reliably identify observations with air leaks and thus have an APU failure and demand maintenance.

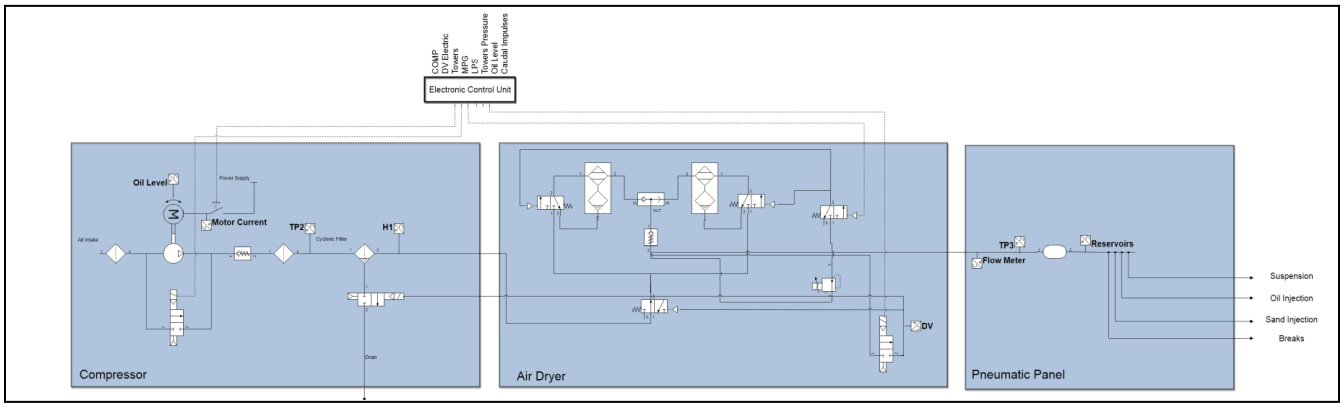
Conclusion

In conclusion, this project aimed to identify observations with air leaks, which is an Air Production Unit (APU) failure. These observations are the minority class, as they initially constitute around 2% of the dataset and thus can be considered anomalies. Accordingly, 4 different approaches were adopted to effectively identify these observations, which are: full data, 2 PCA components on the full data, half the data with the same label ratio, and a balanced dataset. Mainly, the techniques used were a clustering-based approach for anomaly detection, a distance-based approach for anomaly detection, a density-based approach for anomaly detection, and machine learning classifiers. Under each of these methods, further techniques were applied; all the techniques used were evaluated based on their precision, recall, and F1-Score.

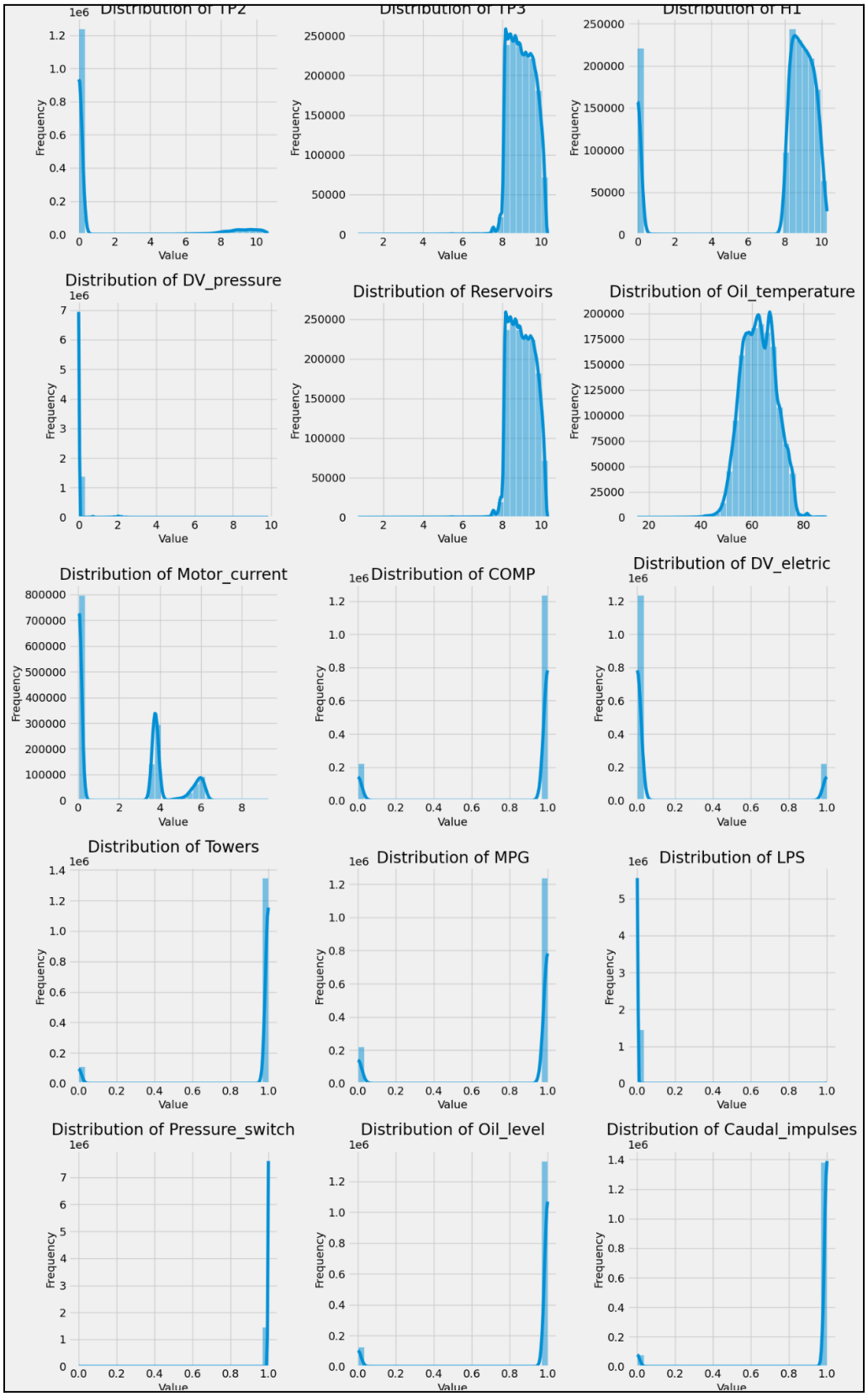
The data was highly challenging due to its very large size and extreme data imbalance, which led to the inability to assess some of the attempted techniques. It can be concluded that machine learning classifiers were the most effective and efficient in achieving the goal of identifying observations with air leaks.

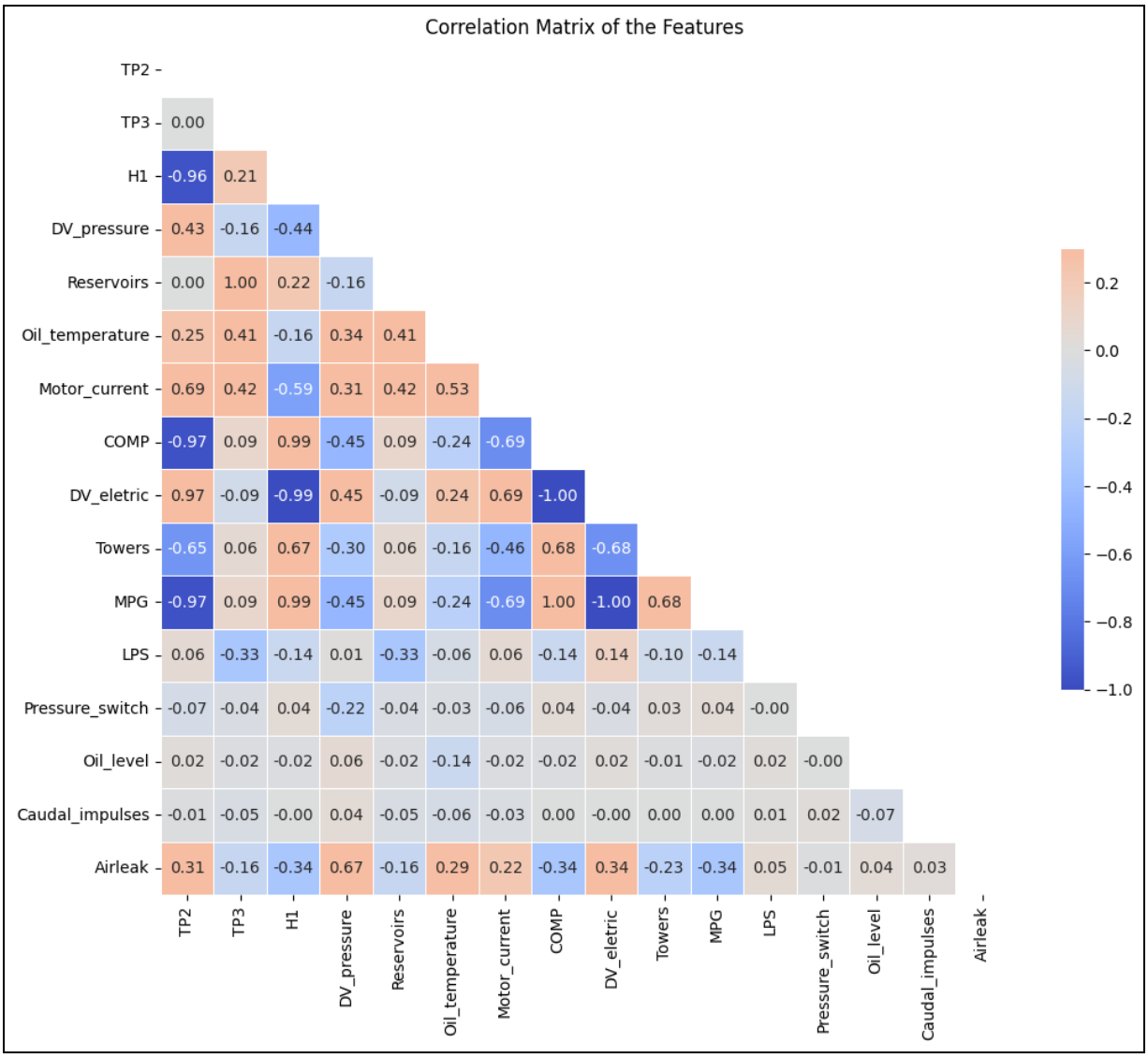
For future recommendations, it is important to evaluate the performance of techniques that failed in our applications, to assess their effectiveness in comparison to the machine learning classifiers. For example, it would be interesting to apply hierarchical clustering and spectral clustering on the dataset to see if they can accurately identify anomalies, yet this would require high computational capacities to be afforded. If this is not possible, a proposed approach is to apply dimensionality reduction using PCA on the sampled data used in approach 3, which is half the original data, and adopt the same anomaly detection approaches to flag observations with air leaks.

Appendix


COMP
DV Electric
Towers
MPG
LPS
Towers Pressure
Oil Level
Caudal Impulses
Electronic Control Unit
Oil Level
Motor Current
TP2
H1
Compressor
Air Dryer
TP3
Flow Meter
Reservoirs
DV
Pneumatic Panel
Suspension
Oil Injection
Sand Injection
Breaks


Visualization of the Original Time Series
TP2
TP3
H1
DV_pressure
Reservoirs
Oil_temperature
Motor_current
COMP
DV_eletric
Towers
MPG
LPS
Pressure_switch
Oil_level
Caudal_impulses
1e6


Distribution of TP2
Distribution of TP3
Distribution of H1
Distribution of DV_pressure
Distribution of Reservoirs
Distribution of Oil_temperature
Distribution of Motor_current
Distribution of COMP
Distribution of DV_eletric
Distribution of Towers
Distribution of MPG
Distribution of LPS
Distribution of Pressure_switch
Distribution of Oil_level
Distribution of Caudal_impulses
Frequency
Value
1e6


Correlation Matrix of the Features
TP2
TP3 0.00
H1 -0.96 0.21
DV_pressure 0.43 -0.16 -0.44
Reservoirs 0.00 1.00 0.22 -0.16
Oil_temperature 0.25 0.41 -0.16 0.34 0.41
Motor_current 0.69 0.42 -0.59 0.31 0.42 0.53
COMP -0.97 0.09 0.99 -0.45 0.09 -0.24 -0.69
DV_eletric 0.97 -0.09 -0.99 0.45 -0.09 0.24 0.69 -1.00
Towers -0.65 0.06 0.67 -0.30 0.06 -0.16 -0.46 0.68 -0.68
MPG -0.97 0.09 0.99 -0.45 0.09 -0.24 -0.69 1.00 -1.00 0.68
LPS 0.06 -0.33 -0.14 0.01 -0.33 -0.06 0.06 -0.14 0.14 -0.10 -0.14
Pressure_switch -0.07 -0.04 0.04 -0.22 -0.04 -0.03 -0.06 0.04 -0.04 0.03 0.04 -0.00
Oil_level 0.02 -0.02 -0.02 0.06 -0.02 -0.14 -0.02 -0.02 0.02 -0.01 -0.02 0.02 -0.00
Caudal_impulses -0.01 -0.05 -0.00 0.04 -0.05 -0.06 -0.03 0.00 -0.00 0.00 0.00 0.01 0.02 -0.07
Airleak 0.31 -0.16 -0.34 0.67 -0.16 0.29 0.22 -0.34 0.34 -0.23 -0.34 0.05 -0.01 0.04 0.03
TP2 TP3 H1 DV_pressure Reservoirs Oil_temperature Motor_current COMP DV_eletric Towers MPG LPS Pressure_switch Oil_level Caudal_impulses Airleak
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0