FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Unsupervised Learning approach for predictive maintenance in power transformers

Duarte Miguel de Novo Faria

Mestrado em Engenharia Informática e Computação

Supervisor: João Moreira

Co-Supervisor: Ricardo Sousa

March 7, 2022

Unsupervised Learning approach for predictive maintenance in power transformers

Duarte Miguel de Novo Faria

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Carlos Soares
Referee: Prof. Nuno Moniz
Referee: Prof. João Moreira
Referee: Prof. Ricardo Sousa

March 7, 2022

Resumo

Os transformadores eléctricos são equipamentos eléctricos fiáveis e eficientes e um dos componentes mais caros de um sistema de energia eléctrica. Devido à carga pesada, a fiabilidade pode variar, motivada pela maximização dos lucros. Assim, é crucial recolher dados a partir deste equipamento para os monitorizar e encontrar anomalias. Contudo, com o aumento dos dados, a análise de base humana tornou-se um desafio e pode ser imprecisa, com a possibilidade de resultar em perdas financeiras e de tempo.

Com isto em mente, o objectivo desta tese é o desenvolvimento de um modelo de machine learning capaz de detectar anomalias num conjunto de dados não etiquetado.

O passo inicial foi agrupar os dados para obter etiquetas. Para isso foram utilizados DBSCAN e K-Means. Depois disto, os dados agora rotulados são equilibrados usando SMOTE e depois submetidos na formação dos modelos de aprendizagem supervisionada, usando alguns algoritmos.

Para os resultados finais, o DBSCAN foi o algoritmo de agrupamento escolhido, e o algoritmo de classificação com o melhor desempenho ROC AUC foi Random Forest com 95,26%. A empresa validou os resultados finais.

Keywords: Predictive maintenance, machine learning, supervised learning, unsupervised learning, power transformers, clustering, DBSCAN

Abstract

Electric power transformers are reliable and efficient electric equipment and one of the most expensive components of an electric power system. Due to heavy load, the reliability can vary, motivated by maximizing profits. Hence, it is crucial to gather data from this equipment to monitor them and find anomalies. However, with the increase of data, human-base analysis became challenging and can be inaccurate, with the possibility of resulting in financial and time losses.

With this in mind, the purpose of this thesis is the development of a machine learning model capable of detecting anomalies in an unlabeled dataset.

The initial step was to cluster the data to obtain labels. For this was used DBSCAN and K-Means. After this, the now labeled data is balanced using SMOTE and then submitted in the training of the supervised learning models by using some algorithms.

For the final results, the DBSCAN was the clustering algorithm chosen, and the classification algorithm with the better ROC AUC performance was Random Forest with 95,26%. The company validated the final results.

Keywords: Predictive maintenance, machine learning, supervised learning, unsupervised learning, power transformers, clustering, DBSCAN

Acknowledgements

First of all, I would like to thank my supervisor, Prof. João Pedro Mendes Moreira, and the co-supervisor, Ricardo Teixeira Sousa, for all the help and advice during the development of this thesis.

I would also like to thank Enging for welcoming me with open arms, especially Diogo Mendes and Jorge Estima, for always being available to help me.

I want to express my gratitude to my family for everything they have done for me. For all the love, for all the sacrifice, and for always being by my side all my life.

Last but not least, I want to thank all my friends who have helped me through this journey. In the happiest and darkest moments, all the sleepless nights, thank you for everything.

To everyone. Thank you!

Duarte Faria

"They're gonna try to tell you no, shatter all your dreams.
But you gotta get up, and go and think of better things."

Mac Miller

Contents

List of Figures

List of Tables

Abreviaturas e Símbolos

ML	Machine Learning
PdM	Predictive Maintenace
CM	Corrective Maintenance
PM	Preventive Maintenance
SVM	Support Vector Machine
DT	Decision Trees
RF	Random Forest
KNN	K-Nearest Neighbors
ANN	Artificial Neural Network
NB	Naive Bayesian
PCA	Principal Component Analysis
TDGA	Total Dissolved Combustible Gas
DGA	Dissolved Combustible Gas
CRISP-DM	Cross-Industry Standard Process for Data Mining

Chapter 1

Introduction

This chapter describes the context in which the project is framed. It aims to explain the motivations around the project, the expected project goals in more depth, and presents a timeline of activities that have been followed since the beginning to the end of the thesis writing. At the end of the chapter, it is made a brief description of the remain chapters.

1.1 Problem description and motivation

Electric power transformers are reliable and efficient electric equipment essential for supplying electric energy to users at appropriate voltage levels. On the other hand, power transformers endure variations in their reliability and operating lifetime over time. These variations in reliability are primarily due to heavy equipment loading. The motivation for this is the need to maximize profits and the electric companies' lack of new equipment investment due to its high cost. However, with enhanced monitoring and maintenance procedures over time, the equipment's lifespan has significantly increased.

This thesis will use data provided by Enging - Make Solutions, S.A., a firm specializing in sophisticated and disruptive industrial predictive maintenance and fault detection solutions for transformers, rotating machines, and power electronics.

Human evaluation of sequential approximation data generated by extracting the equipment's input and output electrical currents is the company's current way of monitoring the state of the equipment. This data contains a considerable number of variables, including the ones listed below that the company has provided, which have a more significant impact on whether or not the equipment is malfunctioning:

- ld-ratio: the transformer load level is represented by this variable.
- fsv-4: this is our target variable, an indicator that translates the excitation currents' amplitude and phase deviation.
- iexc-1 to iexc-3: these indicate the electrical current level in each phase;

- unbi-1 and unbi-2: refers to the amplitude imbalance between the primary and secondary currents.

This human-based analysis can be inaccurate, resulting in severe financial and time losses. As a result, using the gathered data to build a machine learning system can enhance accuracy and cut down on time spent on this work.

1.2 Dissertation objectives

The major goal of this dissertation is to research, develop, and test techniques for diagnosing power transformers using machine learning approaches. The intent is to cluster the gathered data using unsupervised learning methods to detect outliers. After that, classification methods will be employed to predict any anomalies in the testing data. The results must next be evaluated to verify if the outlier corresponds to an anomaly. Because each power transformer has its distinct behavior, it is necessary to define what constitutes an outlier and determine whether it is viable in the real world. Finally, to obtain the most accurate results, it is critical to verify the reliability of the constructed system.

1.3 Dissertation structure

The dissertation is divided into five chapters. Chapter 2 addresses the existing scientific literature that meets this dissertation's context, motivation, and objectives. Chapter 3 proposes and explains the methodology and the tools used to solve the proposed problem. Chapter 4 presents the results obtained by the trained models. Finally, chapter 5 will present the project's conclusion and proposes future work to improve the methodology implemented.

Chapter 2

Literature Review

This chapter discusses the literature review about the maintenance methods for electrical equipment, failure curves in industrial equipment, and the machine learning algorithms.

2.1 Power Transformers

One of the most expensive and strategic components of an electric power system are power transformers. It plays a vital function in the electricity transmission and distribution system by interconnecting all stages. Due to its complex operating conditions under numerous circumstances such as high temperature, emergency overloading, and continuous operation in an outdoor environment, the power transformer is a piece of high-risk equipment in the electric power system [22].

2.1.1 Failures in power transformer

Because this component is such a crucial part of the electric power system, its failure can result in huge losses, not only in replacement or repair losses but also in revenue. Another problem that failures can cause is the reduced reliability of the system over time. It is essential to know its components to understand the possible failures in power transformers. The transformer has a magnetic circuit, electrical circuit terminals, bushings, tank, oil, radiator, conservator, and breathers. A malfunction in any of these parts can jeopardize transformer usability and lead to its power grid withdrawal [16].

2.1.1.1 Failure curves in industrial equipment

There are currently six curves that are considered failures models for industrial equipment. These curves are represented in Figure 2.1, which can be divided into two groups. The first group is time-based maintenance, and the second group is condition-based maintenance[12, 42]

For time-based maintenance, there are 3 types of curves:

- Bathtub: this curve represents an age-related equipment failure. It starts with infant mortality followed by a constant or gradually increasing failure probability and a pronounced

wear-out region. An age limit may be desirable, provided many units survive to the age at which wear-out begins.

- Wear-out: this curve represents a wear-out failure. It starts with a constant or gradually increasing failure probability and then a pronounced wear-out region.
- Fatigue: this curve represents a fatigue failure. This curve is similar to the wear-out curve but shows a gradual increasing failure probability without a pronounced wear-out region.

For condition-based maintenance, there are 3 types of curves:

- Initial break-in period: this curve represents when a new component is installed. In the initial stage, the equipment is not working at total capacity, so the failure probability is lower in this period.
- Random: This curve represents a random failure. It is a constant failure probability curve.
- Infant mortality: this curve represents the accentuated failure probability in an initial period, and then the probability is gradually reduced until it reaches a constant value.

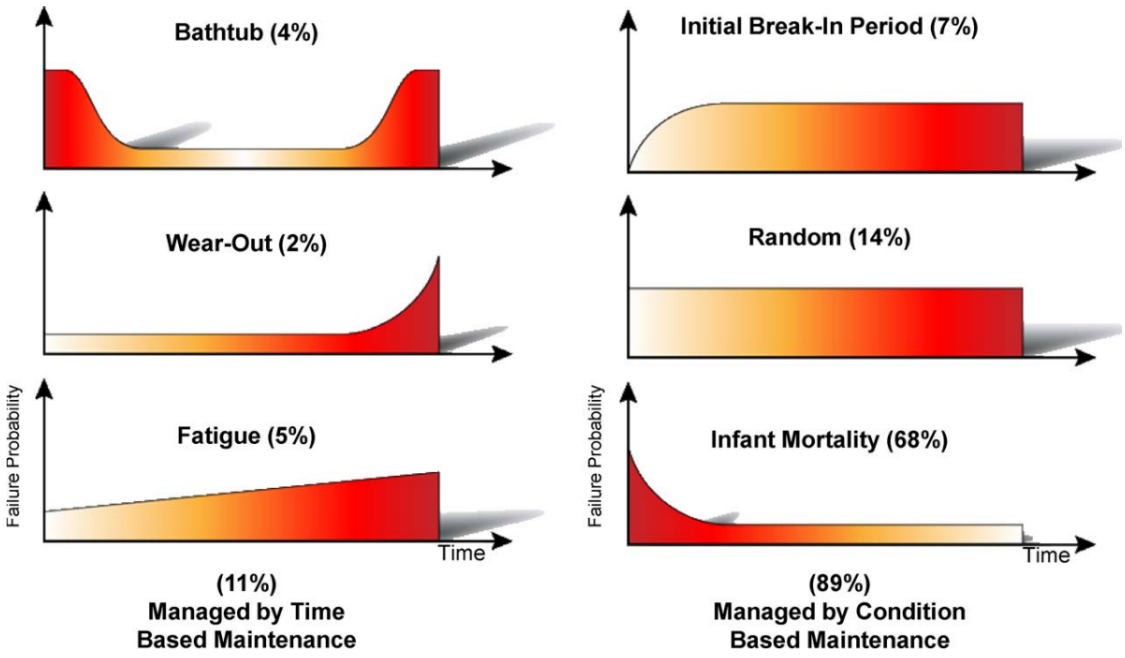

Figure 2.1: Failure curves. Source [12]

2.1.2 Maintenance methods for power transformers

Maintenance is considered a strategic activity that ensures the operation reliability of equipment and industrial processes. Maintenance should seek the intervention in equipment by reducing the

intervention time, leaving the system unavailable for the shortest time possible [6]. Between the various kinds of maintenance, we can highlight three main types, which are:

- Corrective maintenance (CM): refers to any task that is performed in order to restore the equipment. The correction is performed as the faults and failures occur [6];
- Preventive maintenance (PM): the objective of this type of maintenance is to prevent that any possible flaws occur, and seeks to improve the reliability and availability of the equipment. The preventive actions are programmed (e.g., performed periodically) in order to avoid failure (e.g., replace a critical part of the equipment [6];
- Predictive maintenance (PdM): this type of maintenance is an additional preventive and corrective maintenance tool. His function is to collect data of the equipment that we want to examine and with that run a diagnosis and trend analysis in order to seek for potential problems. The actions of maintenance are guided by predictions (predict the occurrence of fault/failure) obtained by associated data [6];

PdM is the one that most relates to the theme of this thesis. Therefore the following subsection will explain the cycle of a PdM system.

2.1.2.1 Predictive maintenance system workflow

In PdM, it is essential to know all the possible types of failures because PdM is based on a decision support system with indicators representing the state of the equipment. Therefore, various types of analysis are done to detect these failures, such as physical-chemical analysis, electrical signature analysis, furfural, particle analysis, thermographic inspection, method of acoustic emission, and dissolved gas analysis, among others.

In order to make predictions about the state of the equipment, a good approach is to use machine learning algorithms. With this said, it is possible to highlight four main steps in a PdM workflow.

The acquiring data phase collects data from the sensors and previously mentioned analysis. The acquiring data phase collects data from the sensors and previously mentioned analysis. After this, it is essential to go through a preprocessing phase to clean the acquired data. Hence, the processed data is passed to a machine-learning algorithm to train a model. Finally, the created model is used to predict anomalies in a new set of data. If some anomaly is detected, the user must be warned to decide if an intervention is needed.

2.1.3 Machine learning applied to PdM

The rising use of data-driven approaches such as machine learning is one of the main reasons for the rise in PdM systems. Maintenance has been transformed due to developments in machine learning. It has been used to anticipate equipment failure, and other relevant events on equipment lifecycle [37].

Depending on the data available, the machine learning approaches employed in predictive maintenance systems may differ. Numerous research articles have been released on this topic. Clustering, classification, regression, and anomaly detection are PdM's most important machine support methodologies. Later in this section, we will go through these strategies in greater depth.

2.2 Machine Learning

With the increase of the data gathered, machine learning techniques became more and more relevant. This subfield of data science is very important because it allows it to extract useful information from the data, which is impossible for an average human to do due to the data size [4].

Machine learning can be divided into four main subfields:

- Supervised Learning:
- Unsupervised Learning:
- Semi-supervised Learning:
- Reinforcement Learning:

For this thesis, the relevant machine learning subfields are supervised learning, unsupervised learning, and semi-supervised learning, which will be discussed later on.

2.2.1 Methodologies

In recent years, data science has gotten much attention, and it has put much effort into developing sophisticated analytics, improving data models, and cultivating new algorithms. However, these projects can face organizational and socio-technical challenges as they progress, such as a lack of vision, strategy, and clear objectives, a biased emphasis on technical issues, a lack of reproducibility, and ambiguity of roles, to name a few. These challenges contribute to a low level of maturity in data science projects that are managed haphazardly [18].

With that said, project management and process methodologies are beneficial to data science projects. Methodologies like these can help succeed and overcome some of the problems mentioned earlier. On the other hand, data science teams may find it challenging to stick to a project methodology. Data science projects can be planned using many methodologies. A survey conducted by KDnuggets in 2014 shows that the most used methodology is CRISP-DM, with 43% of the responders [14].

2.2.1.1 CRISP-DM

CRISP-DM stands for Cross-Industry Standard Process for Data Mining, and it was created in the mid-1990s by SPSS and Teradata. It describes common approaches used by data mining experts. It breaks down into 6 phases of the lifecycle of a data mining project, as can be seen in

the Figure: business understanding, data understanding, data preparation, modeling, evaluation, and deployment [28, 32].

- Business Understanding - This is the first stage, and it is here that the project objectives and needs are defined from a business standpoint. Here is also the time where the project plan is projected.
- Data Understanding - It is critical to analyze and become familiar with the data once it has been collected. The main focus in this phase is to study the data and ensure its quality. This phase is frequently linked to the business understanding phase since it is critical to consider the quality and features of the data in order to ensure that the project objectives are clearly defined.
- Data Preparation - The primary goal of this phase is to handle the data and create the final dataset fed into the modeling. The feature selection, data transformation, and cleaning are conducted in this process.
- Modeling - The modeling techniques are chosen and applied in this phase. The dataset is partitioned, test and train datasets are generated, and models are constructed and applied. This step might also be related to the data preparation phase if some transformation or selection is required to produce better results.
- Evaluation - This is the stage in which the model's output is analyzed and reviewed to see if the defined objectives from the business understanding phase were met or not.
- Deployment - The model is organized and delivered to the customer at this step of the lifecycle.

The usage of machine learning in these types of problems is a widespread practice nowadays.

2.3 Supervised Learning

In this machine learning subfield, the goal is to predict the outcome of a task based on the data's features. The model receives a set of features and a target variable. With this, the model learns the function that relates the features to the target variable [4, 15, 17, 23].

Supervised learning can be divided into two main types:

- Regression: Find a relation model (mathematical function) that relates a set of numerical or categorical variables to one numerical variable;
- Classification: Find a relation model (mathematical function) that relates a set of numerical or categorical variables to one categorical variable;

2.3.1 Supervised Learning Models

Many supervised learning algorithms can be used to solve classification problems. The most relevant ones to this work are Bayesian networks, Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest, K-Nearest Neighbors (KNN), Artificial Neural Network (ANN).

Each algorithm will be described now in its own section.

2.3.1.1 Bayesian networks

A Bayesian Network is a graphical model for probability relationships among a set of variables. The network has to be depicted. Then the parameters are determined, making it challenging to implement without an expert opinion. BN is also not successful with larges datasets because large networks are not feasible in terms of time and space [33].

2.3.1.2 Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem. It relies on the naive assumptions that each pair of features is independent, which means each feature is independent of the other features for a given class. This method is simple, intuitive, and can work with high efficiency [17].

2.3.1.3 Support Vector Machine

Support vector machines are supervised learning algorithms that use a kernel function to transform the features of the data into a high-dimensional space. The goal is to find a hyperplane that separates the data into two classes. The algorithm is based on the idea that the distance between the data points and the hyperplane is the most critical factor in determining the class of the data point [17].

2.3.1.4 Decision Tree and Random Forest

Decision trees are a widely used algorithm for classification problems. They are straightforward to understand and explain once it shows the decision to split the data. This algorithm is robust to the noise in the data. It also provides high performance for relatively fast computation time. One problem with this algorithm is that it finds it difficult to handle high-dimensional datasets. Another problem is that without the proper use of pruning can easily lead to overfitting. In order to solve this problem, random forests are used. A random forest is a collection of decision trees trained using a bootstrap sample of the original data. The goal is to find a set of decision trees that are more robust to the noise in the data [4, 17, 34].

2.3.1.5 K-Nearest Neighbors

K-nearest neighbors (KNN) is a non-parametric method for classification. It is a simple algorithm that can be used to find the nearest neighbors of a given point in a dataset. The goal is to find the class of the nearest neighbors. This algorithm is a simple lazy learning algorithm that largely depends on the value of the k parameter and the size of the data for its efficiency [4, 17, 34].

2.3.1.6 Artificial Neural Network

Artificial neural networks (ANN) are an interconnection between computational models and a layered structure. This network consists of nodes (artificial neurons), weighted connections, and functionality. ANNs idea is to parameterize a structure repetitively to tweak the parameters during training.

The neurons are arranged in layers, and each one is associated with 2 variables, a set of weights and a bias. We can see this structure in Figure 2.2. ANN can be divided into 3 phases, input, processing, and output. The larger the processing phase is, the deeper the neural network is [4, 17, 34].

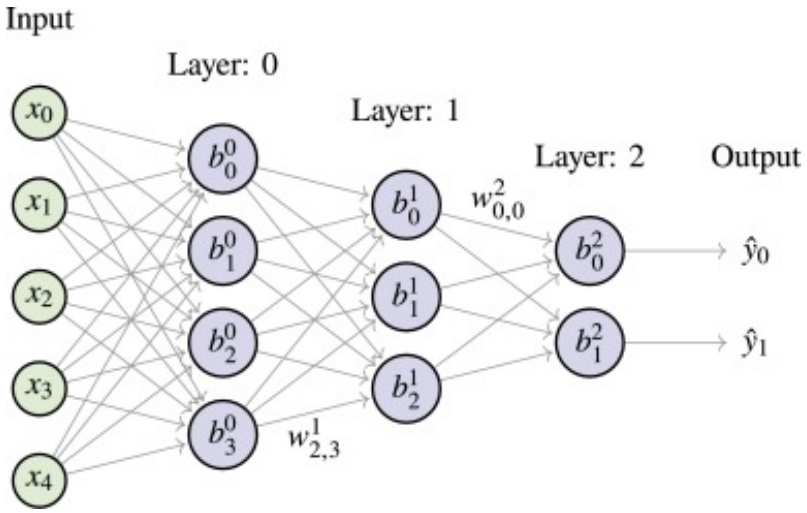

Figure 2.2: An example of an ANN network [4]

2.3.2 Imbalanced data problems

Imbalanced data sets are a common problem in machine learning. This section will discuss some common issues that arise when dealing with imbalanced data sets. It is often to encounter problems with imbalanced datasets in the real world. For example, there is a vast difference between the number of faults and the number of non-faults data points in fault detection.

This can be a problem when training a model to detect faults since the algorithms perform poorly because they are generally designed to handle balanced data sets. One of the most common solutions to handle this type of problem is to use resampling techniques [40, 44, 45].

2.3.3 Overfitting and Underfitting

Overfitting and underfitting are two problems commonly found in machine learning algorithms. Overfitting occurs when the algorithm fits the training data and memorizes its noise. This leads to deterioration of generalization of properties of the model, which results in poor performance when applied to the testing data.

Datasets with small sizes are more prone to overfit than datasets with large sizes, although it can occur in large datasets due to the complexity of the data. On the other hand, underfitting occurs when the algorithm cannot detect the variability of the data. This leads to poor performance in both the training and testing data [1, 2, 30, 31, 43].

In order to solve overfitting and underfitting, several techniques can be used. Some of them are:

- Cross-validation: This technique divides the data into one or more training and testing sets. The goal is to train the model only using the original training set.
- Regularization: this is a technique that penalizes the model to avoid overfitting. This also can be a hyperparameter that can be tuned depending on the algorithm used.
- Feature selection: this is a technique that selects the most relevant features in the data.
- Early stopping: this is a technique that stops the training process when the model is not improving.
- Train with more data: training with more data can be useful to algorithms to detect the signal better. The problem is that if the data added adds noise, this technique is not useful.

The techniques relevant to this work are feature selection and training with more data.

2.3.4 Most common metrics for Classification

It is crucial to evaluate the performance of the model in order to know if it has a good performance and validate it. Many metrics can be used to evaluate the performance of the model. The most common metrics will be described now, each in its section [20, 13].

2.3.4.1 Accuracy

Accuracy is the percentage of correct predictions. It is the most common metric used in machine learning. It is calculated by dividing the number of correct predictions by the total number of predictions. Accuracy is a good metric if the data is balanced. Otherwise, it can give a false sense of achieving high performance.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

Where:

- TP: true positive
- TN: true negative
- FP: false positive
- FN: false negative

2.3.4.2 Precision

Precision is used to measure the conventionally assumed positive patterns correctly predicted from the total predicted patterns in a positive class. Therefore, it is calculated by dividing the number of true positives by the sum of the true positives and the false positives.

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

2.3.4.3 Recall

Recall is used to measure the fraction of positive patterns that are correctly classified. It is calculated by dividing the number of true positives by the sum of the true positives and the true negatives.

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

2.3.4.4 ROC AUC

Area Under Curve (AUC) is one of the most popular metrics that measure the performance of a binary classifier. This metric evaluates the overall performance of a classifier.

AUC ROC is a probability curve that plots the Recall against the False Positive Rate (FPR). This curve is represented in Figure 2.3

$$FPR = \frac{FP}{(TN+FP)}$$

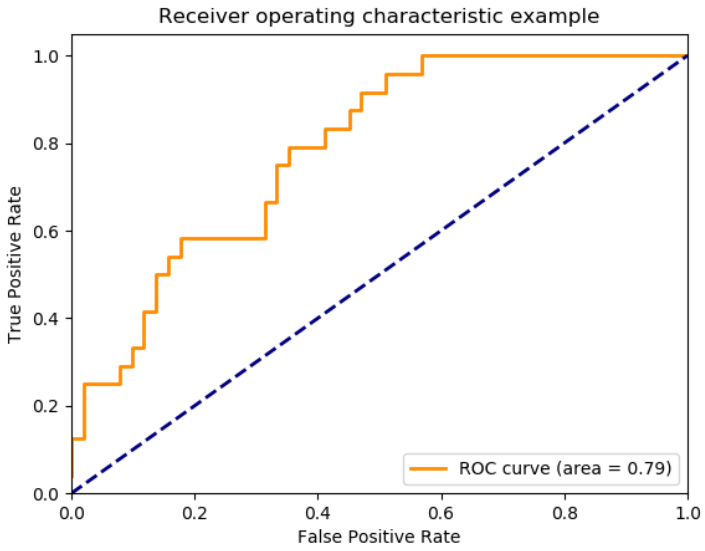

Figure 2.3: Representation of ROC AUC curve [20]

2.3.4.5 F1 Score

F1 score is the harmonic mean of precision and recall. This metric can tell how precise and how robust the model is. F1 score tries to find a balance between precision and recall.

$$F1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

Usually, for binary classification problems, the confusion matrix is the main base for evaluating the model's performance.

The most used metric derived from the confusion matrix is accuracy. However, this metric is not appropriate when dealing with imbalanced data sets. Since the majority class is more impactful than the minority class, which can lead to the model does not perform well to the minority class even if the accuracy is high.

On the other hand, some metrics can be used to evaluate the model's performance when dealing with imbalanced data sets. For example, where can be highlighted, the F1 score, which is a weighted average of the precision and recall, and ROC AUC, which is the area under the ROC curve. Both these measures are not biased towards both majority and minority classes [44].

2.3.5 Related Work

As seen in Table 2.1, the literature presents a vast amount of developed work in this area. The approaches are similar around the papers where the DGA data is used to predict possible failures. The most popular algorithms used are ANN and DT.

Methodology-wise, all the articles used a similar approach. That can be divided into three main parts, data analysis and pre-processing, application of the chosen machine learning algorithms, and finally, comparing the results to understand which one gives the best performance.

Table 2.1: Research papers related to the usage of Classification on predictive maintenance

Transformer failure types	**Algorithms**	**Accuracy (%)**
Irregular dissolved gas concentration [19]	ANN	82
	SVN	80
Leakage and other anomalies[46]	ANN	97.4
	NB	95.9
	DT	95.9
Irregular dissolved gas concentration [7]	DT	85
	SVN	82
Irregular dissolved gas concentration, transformer oil insulation [8]	DT	77
	SVN	70
Irregular dissolved gas concentration[36]	ANN	93.5
Summary on transformer faults[21]	ANN	96.8

In [45] is used a transformer dissolved gas analysis to detect faults in a gas system. The dataset has 9595 healthy data points and 993 faulty data points. This paper compares the performance of the model with various resampling techniques and various classical classifiers with a proposed new approach to the problem named self-paced ensemble (SPE). The results show that the proposed method SPE leads better recall and G-mean than the traditional methods, showing that SPE can deal well with imbalanced data sets.

In [44] is done a review of some popular methods to handle imbalanced data sets. It is used 15 different datasets in this experiment. It is applied undersampling and oversampling techniques and various classifiers to the datasets. F-score and AUC are used to measure the performance of the models in this experiment. The results show that the linear algorithms perform better after applying imbalance techniques in terms of AUC without changing F-score. For the more complex ensemble methods, the application of those techniques does not significantly impact the performance. Lastly, for the simple non-linear algorithms, applying imbalance techniques lacks a consistent performance improvement.

2.4 Unsupervised Learning

Unsupervised learning aims to find patterns and relations among the features of the input data. Hence, unsupervised learning methods work primarily with unlabeled data. Using these methods can help find previously unknown patterns in the dataset [17, 27, 35].

The most common uses of unsupervised learning are:

- Clustering: an approach to finding similar patterns and relations between features in the dataset, grouping the data points with common characteristics[17].

- Association rules: as the name suggests, are simple associations rules to help discover relationships between seemingly independent datasets. This approach is used with large transactional datasets. More specifically, in market basket studies to analyze customers' purchase habits [35].

- Dimensionality reduction: is the process of projecting high-dimensional datasets to a lower-dimensional space. One of the most used methods is the Principal Component Analysis (PCA)[25].

For this thesis, clustering is the methodology explored. The following section presents an overview of the existing clustering algorithms.

2.4.1 Clustering Algorithms

Due to the increase of data and computational power, new algorithms are constantly being developed to answer business needs. With this increase, it is essential to structure a taxonomy to catalog the different algorithms [41].

The most used approach in the literature is to distinguish between partitioning-based, hierarchical, density-based, grid-based, and model-based clustering algorithms.

Partitioning-based algorithms divide the data into a set of k clusters and then assign each data point to the closest cluster. Some examples of partitioning-based algorithms are k-means, k-medoids, fuzzy c-means [41].

Hierarchical algorithms organize the data in a tree-like structure. The tree's root is the first cluster, and the leaves are the closest clusters to the root. Some examples of hierarchical algorithms are BIRCH, CURE, ROCK [41].

Density-based algorithms are used to find clusters in the data close to each other in high-density regions. Some examples of density-based algorithms are DBSCAN, OPTICS, and Mean-Shift [41].

In Grid-based algorithms, the data is divided into a defined grid structure. It quantizes the object areas into a finite number of cells that form a grid structure on which all of the operations for clustering are implemented. The benefit of the method is its quick processing time, which is generally independent of the number of data objects, still dependent on only the multiple cells in each dimension in the quantized space. Some examples of grid-based algorithms are WaveCluster, STING, CLIQUE [41].

Finally, model-based algorithms aim to optimize the fit between the given database and a particular model for each cluster. Some examples of model-based algorithms are EM, COBWEB, SOM [41].

For this thesis were used partitioning-based algorithms and density-based algorithms, in more particular, DBSCAN and k-means.

2.4.1.1 K-Means

K-means is a partitioning-based algorithm that divides the data into k different clusters. Initializing k different centroids, the algorithm then assigns each data point to the closest cluster. The algorithm then recalculates the centroids of each cluster and repeats the process until the centroids do not change. The algorithm is fast and can be used to find patterns in the data [3]. In Figure 2.4 it is possible to see an example of K-Means with k=2 and k=3.

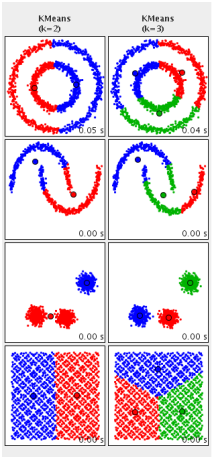

Figure 2.4: Example of K-Means in action with two different k

2.4.1.2 DBSCAN

DBSCAN is a density-based algorithm that finds clusters in the data that are closely packed together and marks the outliers that are not close to any region. The algorithm is straightforward and fast and can be used to find patterns in the data [29, 10]. In Figure 2.5 it is possible to see the performance of DBSCAN using the same datasets in the K-Means example.

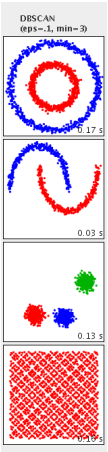

Figure 2.5: Example of DBSCAN in action in the same datasets used in the K-Means example

2.4.2 Metrics for Unsupervised Learning

After using the clustering algorithms, it is essential to evaluate the algorithm's performance. Several aspects are crucial to consider when evaluating the performance of the clustering [11, 24]. These aspects are:

- The clustering tendencies in the data
- The correct number of clusters
- The quality of the clusters without external information
- Comparing the results with external information

Hence, there are two types of validations for clustering algorithms, internal validation, and external validation.

Internal validation is the evaluation of the algorithm's performance without external information, using only the information provided by the input data. There are two types of metrics: cohesion and separation measures. Cohesion evaluates how closely the elements of the same cluster are to each other, while separation measures quantify the level of separation between clusters. Some examples of metrics that measure separation and cohesion at the same time:

- The Calisnki-Harabasz coefficient: also known as the variance ratio criterion, is based on the internal dispersion and the dispersion between clusters [24];
- The Xie-Beni score: this metric was designed for fuzzy clustering. However, it can be applied to hard clustering. It is a ratio that divides the level of compaction of the data within the same cluster and the separation of the data from different clusters [24];
- The Ball-Hall index: The Ball-Hall index: is based on the quadratic distances between the cluster points and the cluster centroid [24];
- Silhouette coefficient: the most common measure that combines cohesion and separation. This measure is defined in the interval [-1, 1] for each data point. In the case of a positive value, a high separation between clusters is experienced. On the other hand, if negative, the clusters are mixed. Finally, if it is zero, it indicates that the dataset is uniformly distributed throughout the euclidean space [24];

External validation is the evaluation of the algorithm's performance with external information. The external validation methods are divided into 3 major groups, which are: matching sets, peer-to-peer correlation, and information theory. The matching sets methods compare the clusters detected with the natural correspondence. Some examples of matching sets methods are:

- Precision: measure the true positives, that is, the number of data points classified adequately within the same cluster;
- Recall: measure the percentage of elements that are adequately included in the same cluster;

- F1 score: the combination of precision and recall;
- Purity: measures whether each cluster contains only samples from the same class;

The peer-to-peer correlation measures the similarity between two partitions under similar conditions, such as a grouping process for the same set. It is assumed that the examples in the same cluster should be in the same class and vice versa. Some examples of peer-to-peer correlation methods are:

- Jaccard coefficient: evaluates the similarity of a detected cluster to a provided partition [24];
- Rand coefficient: is the equivalent to accuracy in a supervised learning approach [24];
- Folkes and Mallows coefficient: calculates the similarity between the clusters found by the algorithm concerning the independent markers [24];

Lastly, the information theory methods are based on Information Theory concepts, such as the current uncertainty in predicting the natural classes provided by other partitions. This family includes basic measures such as entropy and mutual information and their respective normalized variants. Some examples of information theory methods are:

- Entropy: a reciprocal measure of purity that measures the degree of disorder in the clustering [24];
- Mutual information: a metric that measures the reduction in uncertainty about clustering results given prior knowledge [24];

2.4.3 Related work

In [9], it is presented an analysis of the different operating periods of a power transformer through dissolved gas concentration using unsupervised learning methods.

First, Principal Component Analysis (PCA) is applied as a pre-process to represent the data with fewer variables. This gives a compact representation of the data. After that, a k-means classification method is performed to group the operating periods. Lastly, the Total Dissolved Combustible Gas (TDCG) analysis is done.

The results showed that the k-means approach is consistent and can give helpful information about the operating periods.

In [38] it is used data from three sensors from a selective laser melting to build a condition monitoring system with unsupervised learning methods.

The data is analyzed pre-processed, and after that, a clustering method is applied. In this case, k-means is the chosen method. The mean sum of squared distances to centers is used to select the k and the statistical features, also referred to as distortion. The lower this distortion is, the more accurate the number of clusters is.

After analyzing the results, it is possible to identify four distinct clusters, regular operation, protection gas failure, pressure failure, and machine stopped.

These works can relate to this thesis since clustering methods are used to find patterns and group the data points into different clusters. With this, it is possible to identify different anomalies, which is the objective of this thesis.

2.5 Semi-supervised learning

Semi-supervised learning is a branch of machine learning that combines supervised and unsupervised learning techniques. This methodology uses both labeled and unlabeled data to train the model. Typically, it is used to help improve the performance on one of those types of learning tasks by using methods that belong to the other learning task. Most usage of semi-supervised learning is focused on classification. For situations with limited labeled data, semi-supervised classification approaches are helpful because supervised learning techniques are insufficient to solve the problem and are unreliable. This is possible because labeled data can be expensive or difficult to collect. If unlabeled data is sufficient in those circumstances, it can improve model performance. Semi-supervised learning is based on three main assumptions that are the foundation of the most semi-supervised learning algorithms [5, 26, 39, 47].

These assumptions are:

- Smoothness assumption: According to this assumption, if two input points in the input space are close to each other, they are most likely in the same class. This assumption is also common in supervised learning, but the advantage of semi-supervised learning is that it can handle unlabeled data [39].

- Low-density assumption: According to this assumption, the decision border should run through low-density areas rather than high-density areas. The smoothness assumption is strongly related to this assumption [39].

- Manifold assumption: The manifold assumption asserts that the input space is made up of lower-dimensional manifolds on which all data points are located and that data points in the same manifold belong to the same class [39].

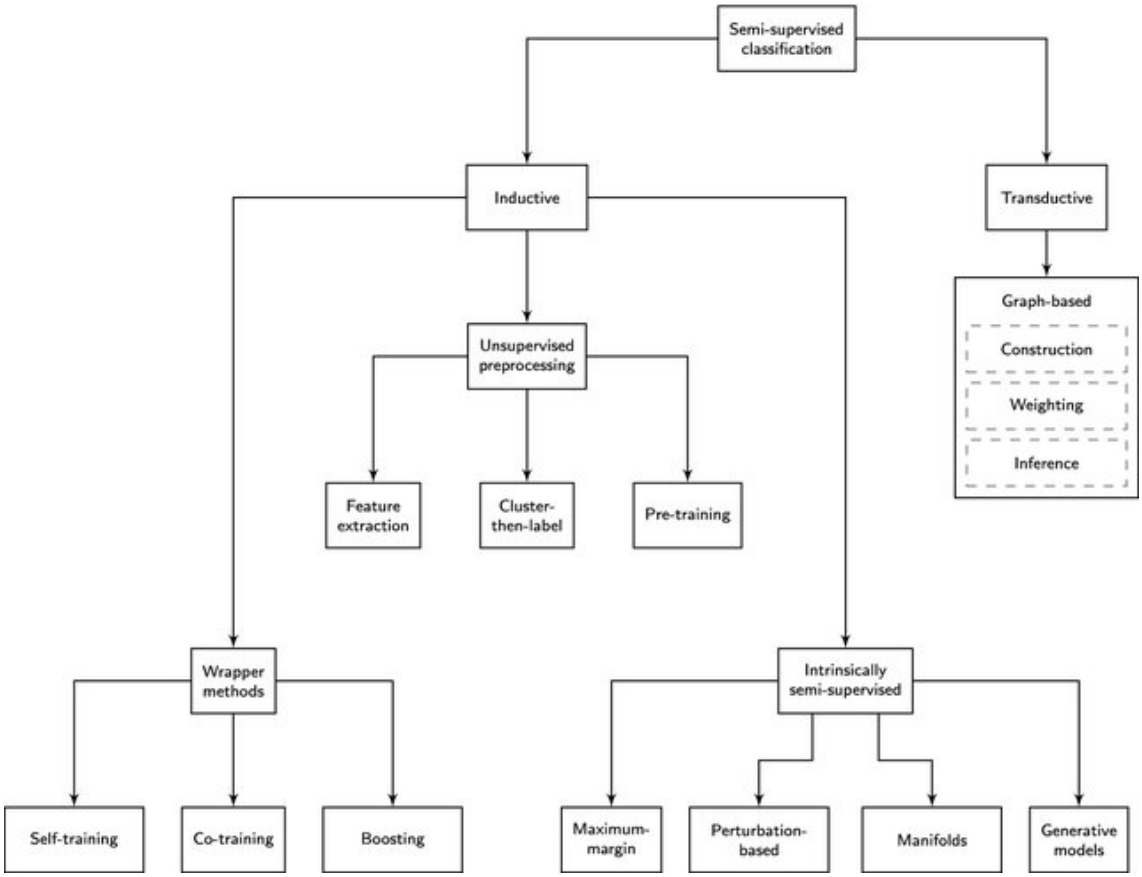

Figure 2.6: Semi-supervised classification taxonomy. Each leaf in the taxonomy corresponds to a specific approach to incorporating unlabelled data into classification methods [39]

In Figure 2.6, semi-supervised taxonomy is depicted. Inductive and transductive approaches are the two primary groups. The following sections will discuss each group and its subgroups in more detail.

2.5.1 Inductive methods

Inductive methods aim to build a model that can create predictions for any type of data available in the input space. These methods are an extension of supervised learning methods to include unlabeled data.

2.5.1.1 Wrapper methods

Wrapper methods train classifiers on labeled data and then produce more labeled data using the predictions. The classifier is then re-trained using the newly labeled data in addition to the previously labeled data. A wrapper approach converts the unlabeled data to labeled data, then is utilized to build the final model using a purely supervised learning algorithm.

Wrapper approaches include the following, which are the most well-known:

- Self-training is the simplest pseudo-labeling method. These methods use a supervised classifier to train iteratively on labeled and pseudo-labeled data until no more data is classified [39].
- Co-training: an extension of self-training in which two or more supervised classifiers are utilized instead of just one. The supervised classifiers must not be too closely connected in their predictions for this to work. If this is the case, the amount of new data generated will be limited [39].
- Boosting: A classifier ensemble is constructed by successively creating individual classifiers, similar to standard boosting algorithms [39].

2.5.1.2 Unsupervised preprocessing

Unsupervised preprocessing uses the labeled and the unlabeled data in two different stages, unlike the wrapper methods. Usually, the unlabeled data is processed first through the unsupervised stage to extract features, cluster the data to label afterward, or pre-train a based learner and initialize with proper weights. The next process is the application of a supervised learning algorithm.

The techniques used in this stage are:

- Feature extraction: This is an advantageous technique, and it has played an essential role in the construction of classifiers. This technique finds transformations in the input data that can improve the classifier's performance or efficiency [39].
- Cluster-then-label: As the name suggests, this approach joins the clustering and the classification processes. First, all the available data is clustered, and then the resulting clusters are used to guide the classification process [39].
- Pre-training: Pre-training is a very used technique nowadays in deep learning. Unlabeled data suggests the boundary towards potentially interesting regions before applying the supervised algorithm [39].

2.5.1.3 Intrinsically semi-supervised methods

Intrinsically semi-supervised methods mainly focus on optimizing the objective function with labeled and unlabeled data components. These methods do not rely on any intermediate step or supervised base learner. These methods generally rely on one of the semi-supervised learning assumptions discussed before.

The methods used in this stage are:

- Maximum-margin: the most straightforward method and attempts to maximize the distance between the input data points and the decision boundary. This technique corresponds to the semi-supervised low-density assumption [39].

- Perturbation-based methods: This technique relies on the smoothness assumption. The predictive model should be robust to local perturbations in the input space, meaning that when a data point is perturbed by a small amount of noise, the prediction to the noisy data and the clean data should be similar. These methods are often implemented with neural networks [39].

- Manifold methods: as seen before with perturbation-based methods, adding small perturbations to the input space works well under the smoothness assumption. However, in some low-dimensional datasets, the perturbations may differ from the input data. For this reason, here is used the manifold assumption [39].

- Generative models: the methods above are all discriminative. Their only goal is to assume a function that can classify data points. In contrast, generative models' goal is to model the distribution p(x, y), from which samples(x, y) can be drawn [39].

2.5.2 Transductive methods

Unlike the inductive methods discussed before, which produce a predictor that can operate over the entire input space, the transductive techniques cannot distinguish between training and testing phases. Labeled and unlabeled data are provided as the input, and the output is exclusively predictions for the unlabeled data [39].

These methods are usually defined by a graph over all data points, encoding the pairwise relationships between data points with possibly weighted edges [47]. It is defined and optimized as an objective function that ensures that the predicted labels match the actual labels for labeled data and that similar data points represented by the similarity graph should have the exact label predictions.

There is a similarity between transductive and inductive manifold methods since both build a graph over the data points and use pairwise relationships to approximate more complex structures. The main difference is that inductive methods aim to create a classifier capable of operating over the entire input space, while transductive methods only give predictions for a given unlabeled data.

2.5.3 Related work

In [5] is proposed three semi-supervised learning algorithms:

- Deriving graph-based distances that emphasize low-density regions between clusters followed by an SVM classifier (graph).

- Optimizing the Transductive SVM objective function by gradient descent ($\nabla TSVM$).

- The combination of the two previous algorithms (LDS).

This experiment used two artificial datasets and three real-world datasets with different properties. For the experience, besides the proposed algorithms, methods from literature and other

research groups were used. The results showed that LSD achieves lower test errors than the other algorithms.

This research paper is helpful because it is proposed a semi-supervised classification algorithm based on cluster assumptions. This type of algorithm could be used in a future iteration of this project.

Chapter 3

Development Work

In this section, it will be presented the development of the project. First, the data set and each feature will be discussed and essential for developing the model. Then, the approach taken to solve the fact that the data provided does not have labels will be presented. Finally, the classification models will be presented.

3.1 Technologies

For this project, we used Jupyter Notebook, a web-based environment for interactive computing in Python. It ables the creating and sharing of live code, equations, visualizations, and narrative text. In terms of python libraries, it was used the following pandas, NumPy, matplotlib, sklearn.

Pandas is a powerful, flexible, open-source library for data manipulation and analysis. Numpy is an open-source library that aims to enable numerical computing. Matplotlib is a Python 2D plotting library with a focus on interactive visualization. Sklearn is an open-source library with simple and efficient data mining and analysis tools.

3.2 Project overview

The project will be divided into two main phases: the clustering and classification phase.

In the clustering phase, the goal is to find patterns in the data and understand what features presents the most impact on the model's decision. First, data understanding is done, which is the process of understanding the data and its structure. Then, the clustering is performed. After the clustering, the data points are labeled with the label of the same cluster (arbitrary label), and it is submitted to a decision tree algorithm to know the impact of each feature in the decision of the model.

After this, it is performed the classification phase. In this phase, the data labeled with the clusters can be treated as a classification problem. The classification problem is solved using various algorithms, and the best algorithm is chosen. For this, some techniques are used, such as

splitting the data into training and test sets, oversampling and undersampling the data, and using different metrics to evaluate the algorithm's performance.

3.3 Dataset

It is crucial to be familiar with the dataset used in a project to make the right decisions about the development of the model.

3.3.1 Data understanding

The data provided by the company was already cleaned, so it was already without missing values. However, it was necessary to clean some existing noise. The company provided three datasets, two from the same device and one from a different one. The first dataset was a three-mouth dataset with 9864 entries and 7 features. The second dataset was from the same machine and had 21838 entries and 8 features. The last dataset was from a different device and had 31087 entries and 8 features.

The goal is to make a model that can work with the first and the second dataset and then later try the same model to the third dataset and see if it can be used to predict the labels.

All datasets have these 7 features:

- ld-ratio: Refers to the percentage of transformer charge;
- fsv-4: Is an artificial feature that is created by the company to try to diagnose the state of the machine;
- iexc-1 to iexc-3: indicates the electrical current level in each phase;
- unbi-1 and unbi-2: refers to the amplitude imbalance between the primary and secondary currents.

The first dataset was used to train the model, so from now on, the dataset will be named as the training dataset. The second dataset will be directed to as the validation dataset. Lastly, the third dataset will be referred to as the validation dataset.

Figure 3.1 shows the scatterplots that reveal the relationship between the features of the training dataset.

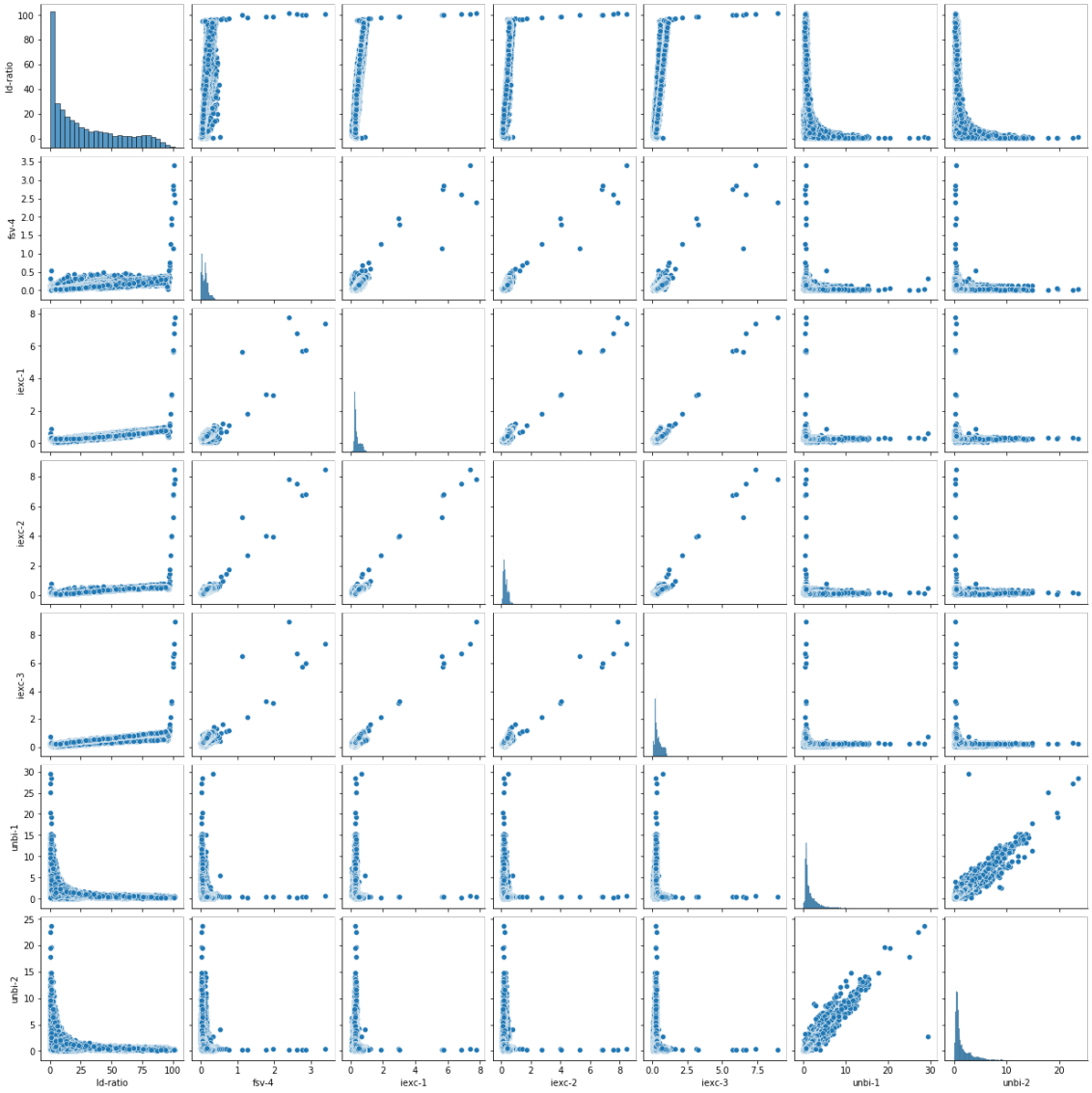

Figure 3.1: Relation between training dataset's features

To a better look at the outliers, Figure 3.2 shows the relationship between the features but with a simple binning. This binning is done by getting the minimum and maximum value of fsv-4 and dividing the range into two bins, 0 and 1.

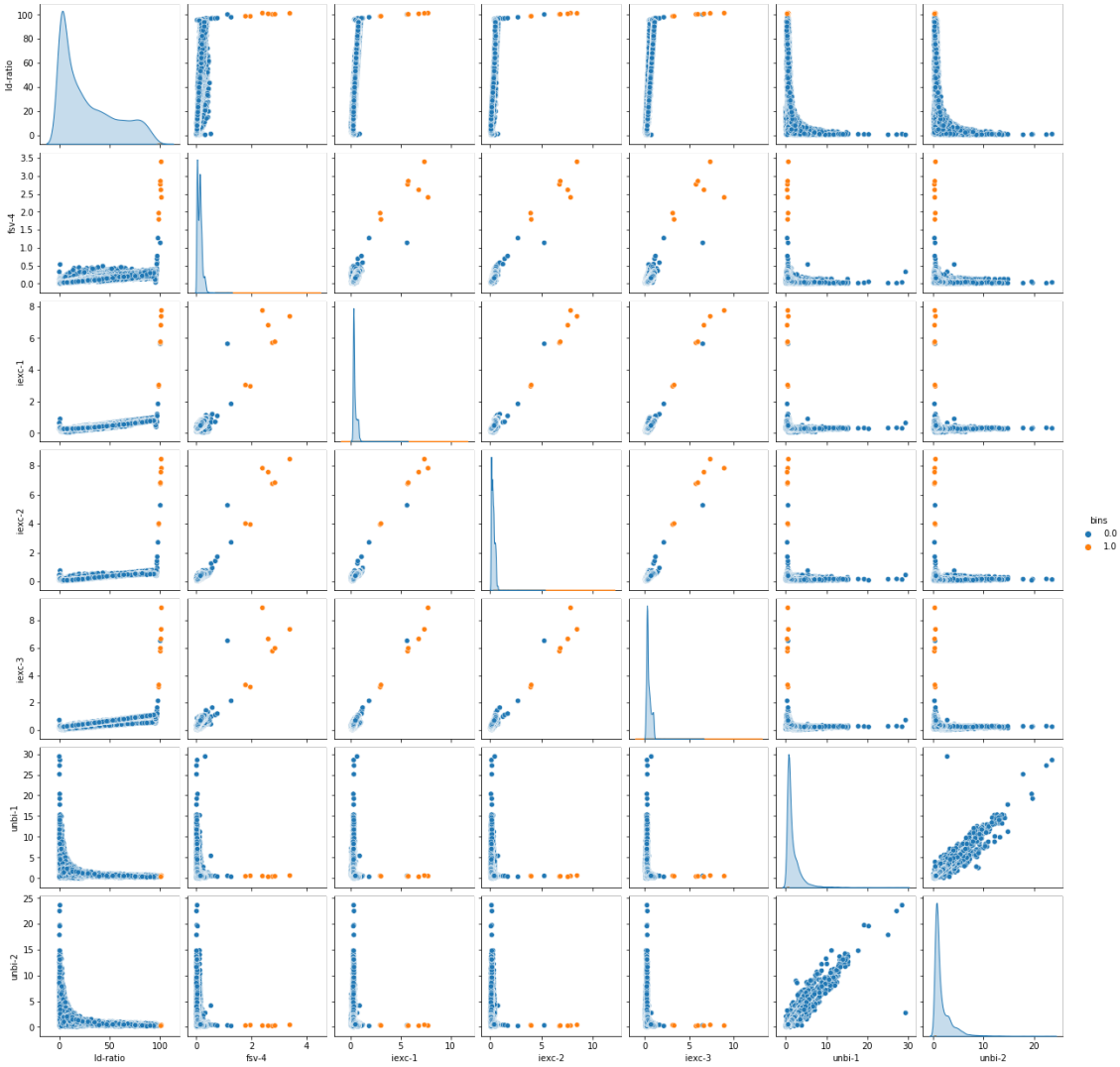

Figure 3.2: Relation between training dataset's features with binning

As can be seen, it is possible to see some data points that can be tagged as outliers. To better understand the relationship between the features, it is used three types of correlations between them:

- The Pearson correlation coefficient measures the linear correlation between two variables.
- The Spearman correlation coefficient measures the monotonic relationship between two variables.
- The Kendall correlation coefficient measures the concordance between two variables.

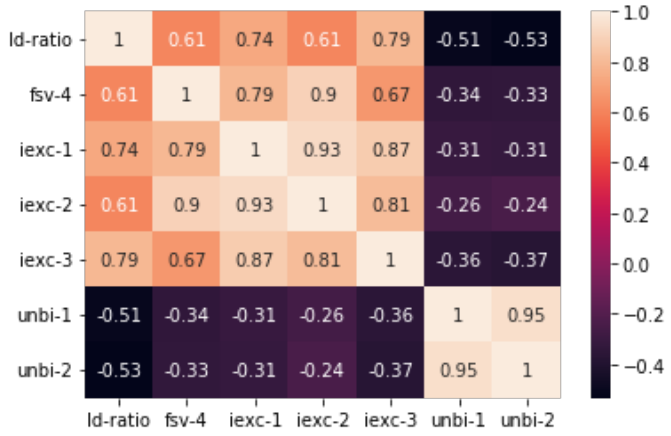

Figure 3.3: Correlation between features using Pearson method.

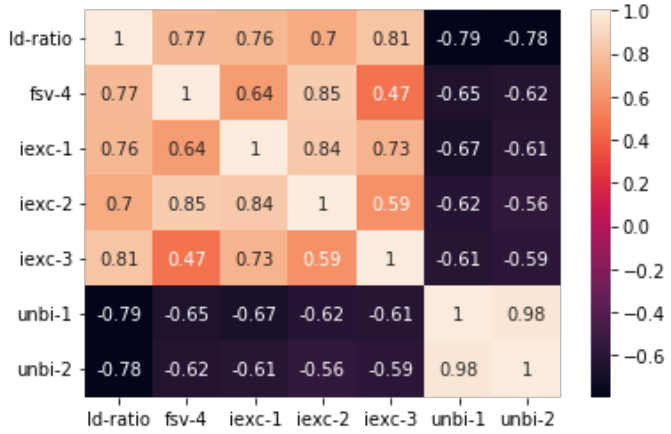

Figure 3.4: Correlation between features using Spearman method.

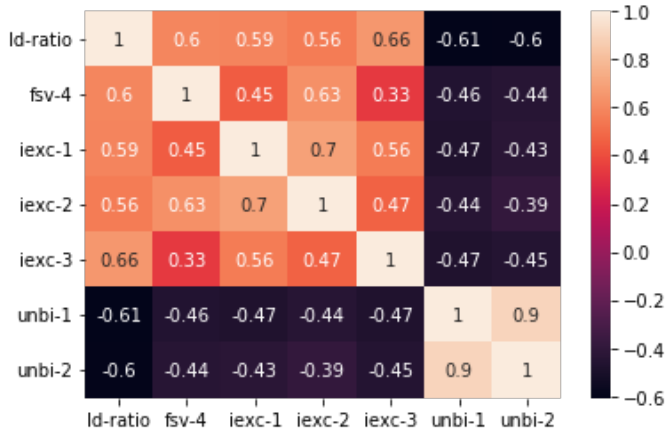

Figure 3.5: Correlation between features using Kendall method.

As can be seen, both unbi-1 and unbi-2 have a low correlation with the others features. The rest of the features presents from moderated to high correlation with each other. The same observation is found in the Pearson, Spearman and Kendall correlation.

3.4 Modeling

Modeling in machine learning is the process of feeding data into an algorithm to detect patterns in the dataset. After this, new data is presented, and the trained model may predict the labels of this new data.

This project was divided into two phases. The first phase used a clustering model to generate labels for the training dataset. The second phase used a classification model to predict the labels of the validation dataset.

3.4.1 Splitting data

Splitting data is essential in model development since it allows evaluating the model without using the validation dataset. The training dataset is split into training and testing datasets. This also prevents the model from overfitting.

The training dataset was split into two parts for the classification model, one for training and one for testing. The data was divided into 70% training and 30% testing datasets.

3.4.2 Feature engineering

Feature engineering is the process of transforming the data into a form that can be useful for machine learning algorithms. The datasets provided by the company were already cleaned, so it is not necessary to do it again. Although it was created one new feature, this feature is a binary that indicates if the fsv-4 is higher than the mean value of the fsv-4.

Another feature engineering method used in the training dataset was normalization, once the feature's range varies massively between them. In this case, it was used the z-norm.

3.4.3 Overfitting and Underfitting

Oversampling and undersampling are techniques used to balance the dataset to improve the model's veracity. In case of oversampling, random instances from the minority class are duplicated to balance the dataset. On the other hand, in undersampling, random samples from the majority class are deleted to balance the dataset.

In this case, the techniques used during the development were SMOTE to deal with oversampling, RandomUnderSampler to deal with undersampling, and TomekLinks to deal with both. SMOTE is a helpful technique since instead of duplicating instances from the minority class. It generates new ones. With this, the problem of adding redundant data is solved.

3.4.4 Optuna

It is essential to use the best hyperparameters of a model to achieve the best performances possible. In this project, the best hyperparameters were found using Optuna.

This library was used to optimize the hyperparameters of the classification model.

3.4.5 Clustering

Clustering is a machine learning technique used to group the data into clusters to make a better analysis of the data and possibly create labels for the unlabeled data.

For this project, it is used two clustering algorithms: K-Means and DBSCAN.

To choose the hyperparameters of the DBSCAN algorithm, a grid search was done.

The grid search was done using the parameters:

- eps: The eps parameter is the maximum distance between two samples for them to be considered in the same neighborhood.
- min_samples: The min_samples parameter is the number of samples (or total weight) in a neighborhood for a point to be considered a core point. This includes the point itself.

	5	10	15	20
0.01	(57, 2271)	(14, 2944)	(8, 3248)	(8, 3410)
0.02	(36, 1027)	(17, 1479)	(17, 1837)	(7, 2229)
0.03	(19, 553)	(7, 839)	(4, 1044)	(9, 1169)
0.04	(8, 303)	(5, 480)	(6, 597)	(4, 696)
0.05	(2, 204)	(2, 286)	(5, 368)	(3, 437)
0.06	(3, 131)	(2, 178)	(3, 220)	(2, 272)
0.07	(2, 97)	(2, 134)	(2, 156)	(2, 179)
0.08	(5, 59)	(2, 109)	(2, 128)	(2, 143)
0.09	(2, 43)	(2, 76)	(2, 101)	(2, 118)
0.10	(2, 34)	(2, 57)	(2, 77)	(2, 89)
0.11	(2, 30)	(2, 43)	(2, 57)	(2, 68)
0.12	(2, 21)	(2, 30)	(2, 38)	(2, 44)
0.13	(2, 15)	(2, 28)	(2, 32)	(2, 38)
0.14	(2, 11)	(2, 19)	(2, 26)	(2, 34)
0.15	(2, 6)	(2, 13)	(2, 15)	(2, 22)
0.16	(2, 3)	(2, 10)	(2, 12)	(2, 16)
0.17	(2, 2)	(2, 7)	(2, 11)	(2, 12)
0.18	(2, 1)	(2, 3)	(2, 7)	(2, 9)
0.19	(2, 1)	(2, 2)	(2, 3)	(2, 6)
0.20	(2, 1)	(2, 2)	(2, 3)	(2, 4)
0.21	(2, 1)	(2, 1)	(2, 1)	(2, 2)
0.22	(2, 1)	(2, 1)	(2, 1)	(2, 1)
0.23	(1, 9854)	(1, 9854)	(1, 9854)	(1, 9854)
0.24	(1, 9854)	(1, 9854)	(1, 9854)	(1, 9854)

Figure 3.6: Grid search to find the best DBSCAN hyperparameters. In the first column, there are the eps, and the first row is min_samples. In each cell, the number of clusters and outliers produced by the pair is shown.

In Figure 3.6, it is possible to see an example of a grid search for the DBSCAN model.

3.4.6 Classification

Classification is a machine learning technique used to learn the relationship between the features and the labels.

In this project, the classification algorithms used were:

- Random Forest (Described in section: 2.3.1.4);
- Decision Tree (Described in section: 2.3.1.4);
- SVM (Described in section: 2.3.1.3);
- KNN (Described in section: 2.3.1.5).

3.4.7 Metrics

In terms of metrics, in 2.3.4 and 2.4.2 it was done a explanation of various metrics used in machine learning. This project used the Silhouette score for the clustering model and the F1 score, recall, precision, and ROC AUC for the classification model.

Chapter 4

Results

In this section, it will be presented and discussed the results of the proposed solution.

This section is divided into two subsections, clustering and classification. The first subsection presents the results from both clustering algorithms used, namely KMeans and DBSCAN. The second subsection presents the results from the classification algorithm used, namely Random Forest, decision tree, and SVM.

4.1 Feature selection

During the development of the project, various tests were performed, and after the first feedback from the company, it was realized that some features were not necessary for the clustering.

This was decided because there were too many anomalies in the predictions with all the features.

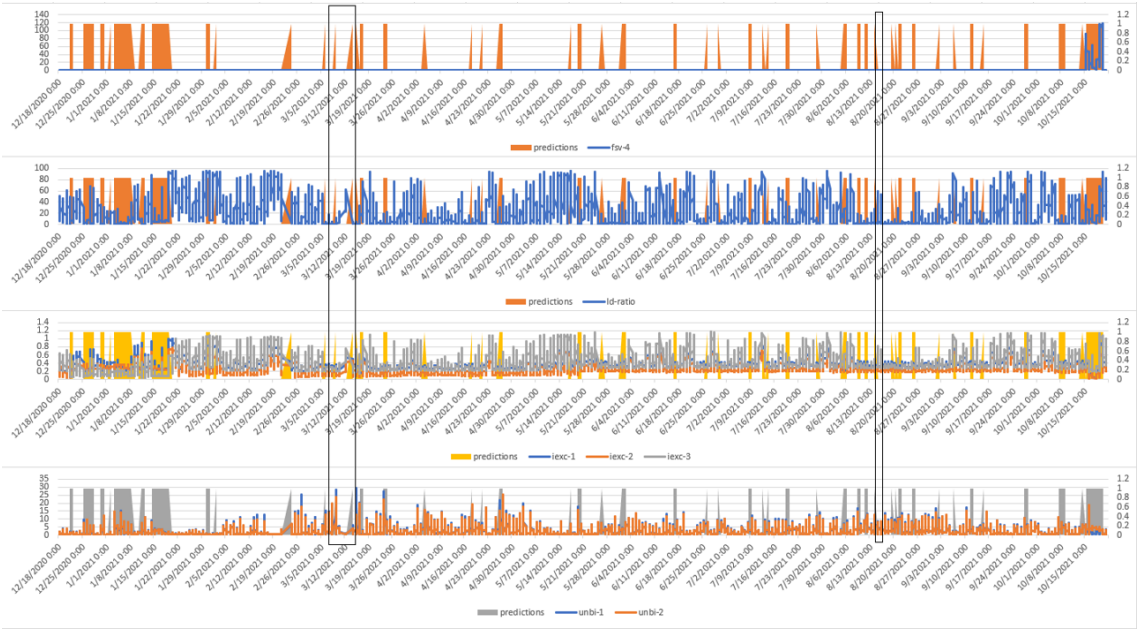

Figure 4.1: Here it is presented the graphs of the number of anomalies predicted per day and the representation of each feature. The anomalies were expected to appear in the highlighted zone. As can be seen, a considerable number of anomalies were predicted outside that zone.

As shown in Figure 4.1, there were predicted anomalies during all year, which does not correspond with the expected results. The anomalies were supposed to be predicted only in the highlighted zones. This happened because some features had too much weight in creating the clusters.

With this said, it was selected three features for the rest of the development, fsv-4, ld-ratio and fsv-4_higher_than_mean.

4.2 First phase

Initially, all the features were used for the clustering because we had little information about the dataset and the expected results. The leading information that we had was that there were some anomalies in the data in the 10, 12, and 16 of March and on the 5 of August of the validation data. We did not know the reason for the anomalies, and we did not know in which instances the anomalies were.

The goal here was to find the best way to cluster the data to find the outliers, using DBSCAN and KMeans.

4.2.1 DBSCAN

Here we used the DBSCAN algorithm to cluster the data. The parameters that were used were the epsilon and the min_samples.

Table 4.1: Grid search for DBSCAN with only ld-ratio, fsv-4 and fsv-4_higher_than_mean as features

		min_sample			
		5	10	15	20
eps	0.1	(8, 132)	(8, 262)	(6, 504)	(7, 758)
	0.2	(7, 38)	(5, 69)	(4, 98)	(3, 130)
	0.3	(5, 20)	(4, 56)	(4, 62)	(4, 72)
	0.4	(3, 16)	(3, 22)	(4, 48)	(4, 49)
	0.5	(3, 14)	(3, 15)	(3, 18)	(3, 27)
	0.6	(3, 14)	(3, 14)	(3, 15)	(3, 17)
	0.7	(3, 14)	(3, 14)	(3, 14)	(3, 14)

Table 4.2: Grid search for DBSCAN with all the features

		min_sample			
		5	10	15	20
eps	0.1	(108, 4394)	(42, 6428)	(31, 7914)	(19, 8959)
	0.2	(37, 878)	(16, 1457)	(17, 1899)	(12, 2289)
	0.3	(20, 368)	(11, 574)	(9, 782)	(10, 930)
	0.4	(13, 184)	(6, 313)	(6, 414)	(5, 505)
	0.5	(8, 93)	(5, 195)	(4, 265)	(4, 300)
	0.6	(5, 54)	(6, 98)	(5, 163)	(4, 210)
	0.7	(5, 43)	(5, 56)	(4, 111)	(4, 124)
	0.8	(5, 38)	(5, 46)	(4, 82)	(4, 96)
	0.9	(5, 37)	(5, 39)	(4, 60)	(4, 79)

As can be seen in Table 4.1 and Table 4.2, each resulting cell gives a pair. This pair contains the number of created clusters and the number of outliers detected.

The criteria to choose the best eps/min_sample pair was to select the cell with the fewer clusters and outliers detected.

4.2.1.1 Using only ld-ratio, fsv-4 and fsv-4_higher_than_mean as features

In this case with only ld-ratio, fsv-4 and fsv-4_higher_than_mean as features the chosen pair was eps = 0.5 and min_sample = 5.

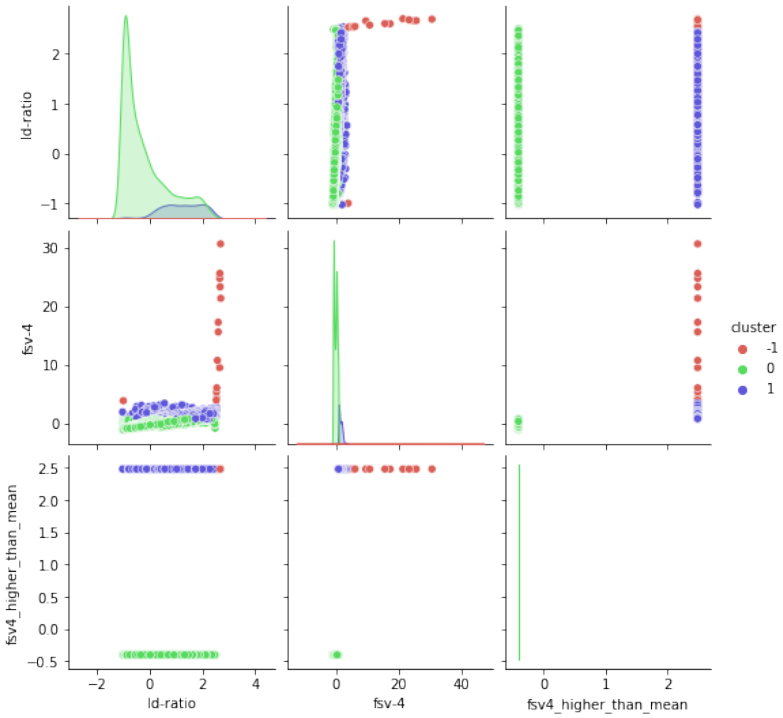

Figure 4.2: Relation with only ld-ratio, fsv-4 and fsv-4_higher_than_mean as features and with the clusters signalized

As it is shown in Figure 4.2, there is a clear distinction between the clusters. Mainly between cluster -1 and the others.

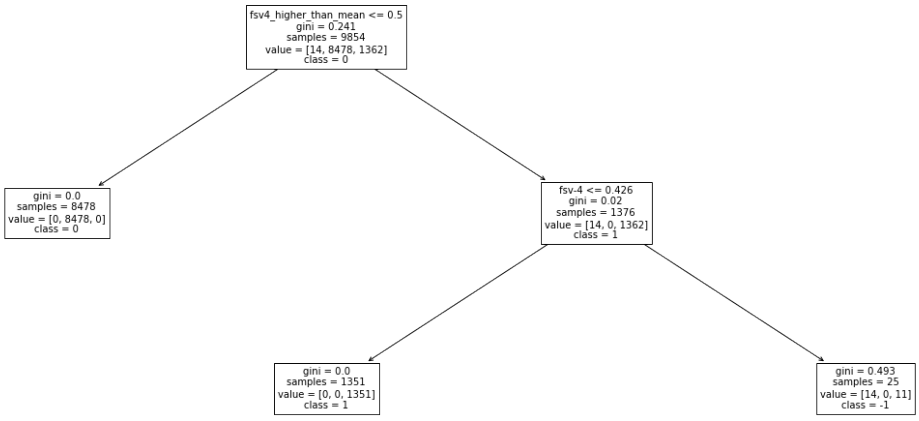

Figure 4.3: Decision tree with the decisions made by the clustering

In this decision tree 4.3, it can be seen the decisions made to divide the data into the clusters. When the fsv-4 is below 0.2, every instance is labeled cluster 0. If the fsv-4 is greater than 0.426, the instances are labeled cluster -1. And finally, the rest are labeled as cluster 1.

4.2.1.2 Using all features

In this case with all features the chosen pair was eps = 0.9 and min_sample = 15.

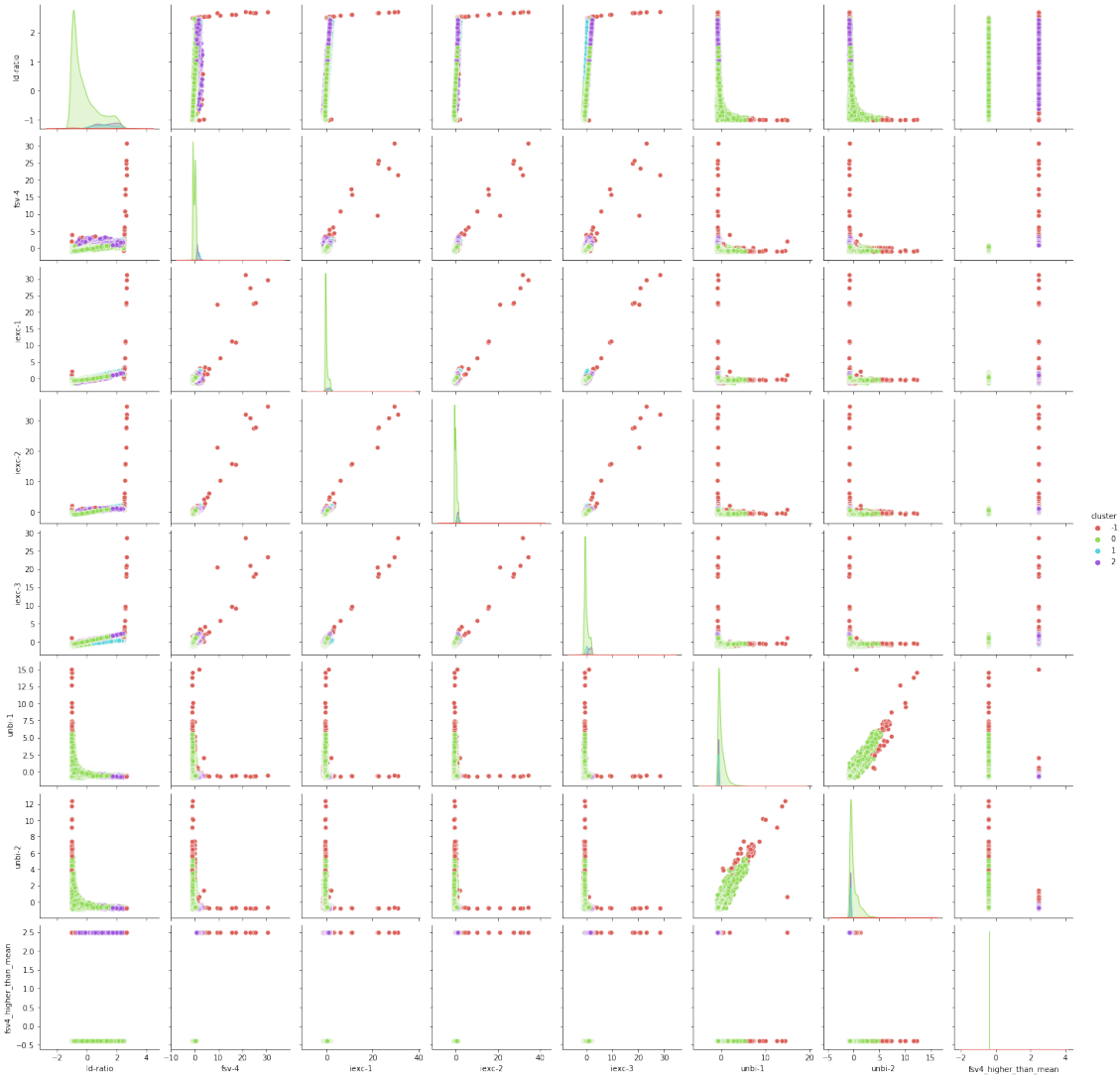

Figure 4.4: Relation with the all features and with the clusters signalized

As it is shown in Figure 4.4, there is a clear distinction between cluster -1 and the others. However, the other clusters overlap, making it difficult to distinguish them.

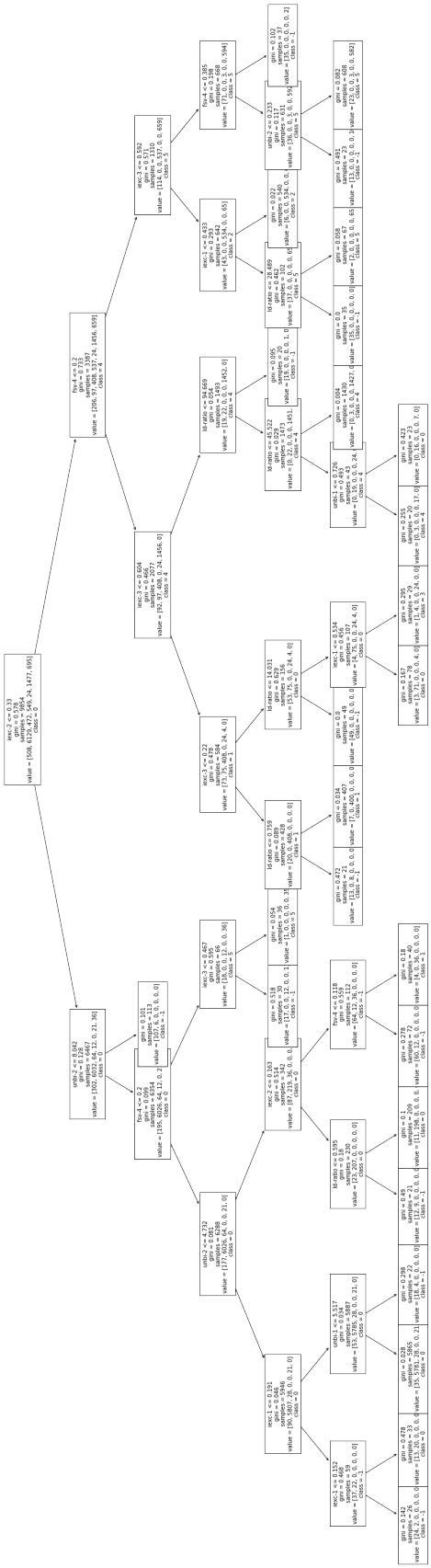

Figure 4.5: Decision tree with the decisions made by the clustering

In Figure 4.5, it is possible to see that it is challenging to understand and to get some knowledge about the decisions made.

4.2.2 K-Means

Here we used the K-Means algorithm to cluster the data. The parameters that were used were the number of clusters.

4.2.2.1 Using only ld-ratio, fsv-4 and fsv-4_higher_than_mean as features

The number of clusters used was calculated by the elbow method. The elbow method is a heuristic to determine the number of clusters in a dataset. A graph with K-Means inertia against the number of clusters created is plotted.

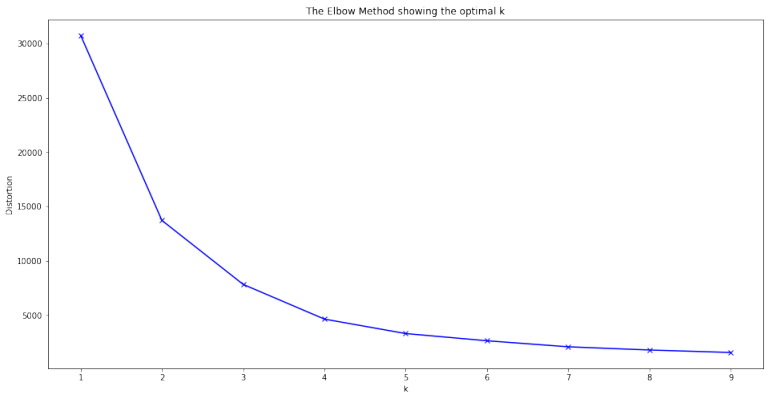

Figure 4.6: The Elbow Method showing the optimal k

As seen in Figure 4.6, the elbow method is used to determine the optimal k. After an analysis, it is possible com conclude that k=3 is the optimal k for these settings.

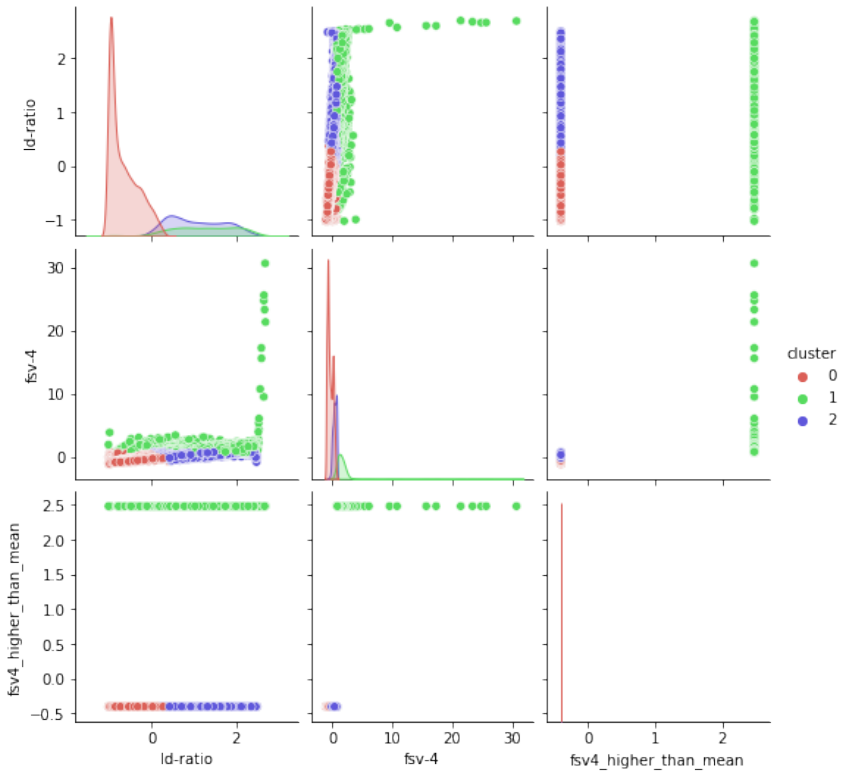

Figure 4.7: Clusters created by K-Means with k=3

In this case, as can be seen in Figure 4.7, it is possible to identify the 3 clusters clearly. The problem with these results is that the outliers are not detected expectedly.

4.2.2.2 Using all features

The number of clusters used was calculated by the elbow method.

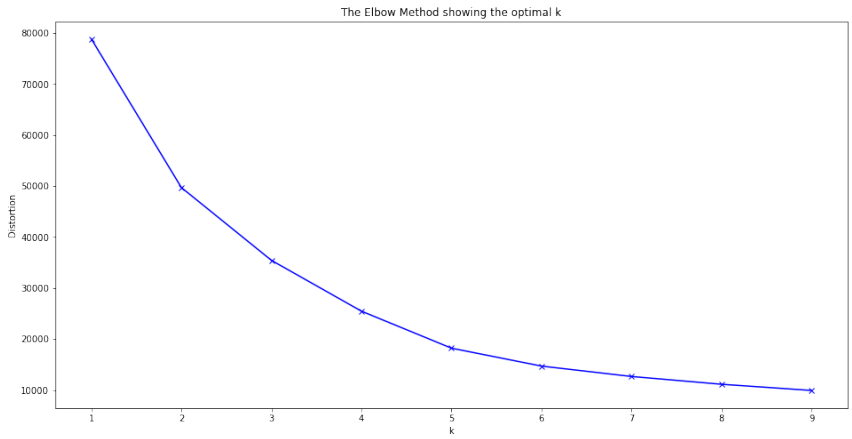

Figure 4.8: The Elbow Method showing the optimal k

As seen in the Figure 4.8 the elbow method is used to determine the optimal k. After an analysis, it is possible com conclude that k=4 is the optimal k for these settings.

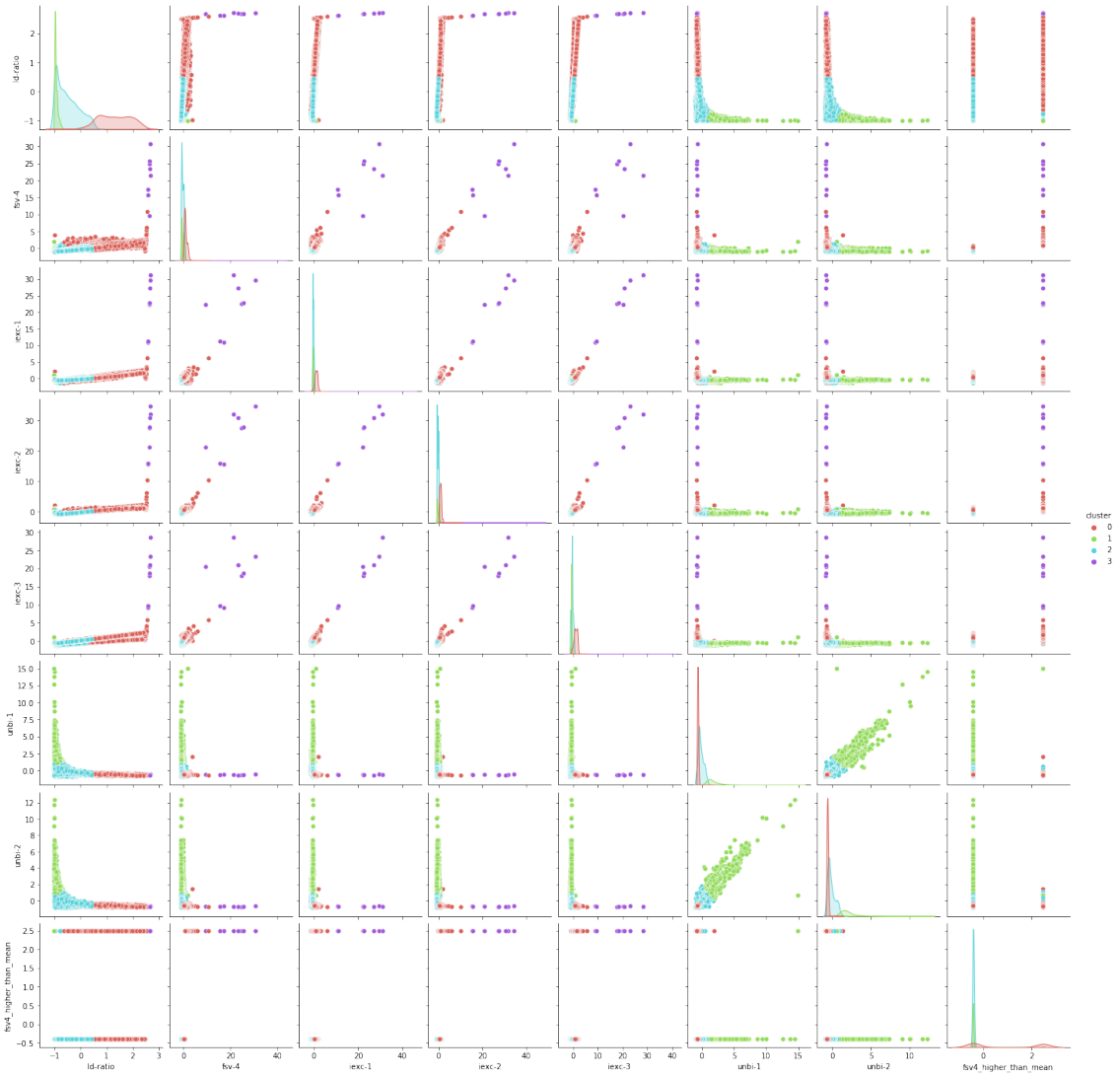

Figure 4.9: Clusters created by K-Means with k=4

In this case, as can be seen in Figure 4.9, it is possible to identify the 4 generated clusters clearly. Here it is possible to identify some outliers grouped in cluster 3. However, this clustering is not as good as the one performed by the DBSCAN.

4.2.3 First phase conclusions

With the obtained results from the DBSCAN and K-Means clustering, it is possible to conclude that the best approach is to use DBSCAN. The clustering using DBSCAN proved to identify the outliers more effectively, and it is possible to observe that this is a density-based problem.

Using only ld-ratio, fsv-4 and fsv-4_higher_than_mean as features clearly distinguishes clusters and gives us a simple decision tree to get knowledge about the decisions made.

In summary, the features used were ld-ratio, fsv-4 and fsv-4_higher_than_mean, and it was used DBSCAN with eps = 0.5 and min_sample = 5 to create the clusters and find the outliers. These clusters are used to label the training data to perform classification.

4.3 Second phase

After the first phase is completed, the now labeled data is used to classify the validation data. To make it easier for the classifiers, the labels will change for 1 if they are an outlier and 0 if not. So, now the training dataset has only two labels, which can be treated as a binary classification problem.

One problem that the dataset has is that the labels are imbalanced. There were only 49 data points labeled as outliers (1) and 9805 labeled as normal behavior (0). To solve this was used techniques to overcome this problem. It was used oversampling, undersampling, and both. For the oversampling, SMOTE was used, for the undersampling was used RandomUnderSampler, and for both was used TomekLinks. After this process, the data is submitted to the classification model. For the classification, it was used, Random Forest, KNN, SVM, and Decision Trees.

Table 4.3: ROC AUC Performance for the chosen algorithms

	Random Forest	Decision trees	KNN	SVM
over	0.9526	0.8791	0.8358	0.7625
under	0.9478	0.8871	0.7886	0.7346
tomek	0.8881	0.8636	0.6626	0.5673

In Table 4.3 it is possible to see that the best ROC AUC performance was using Random Forest and oversampling.

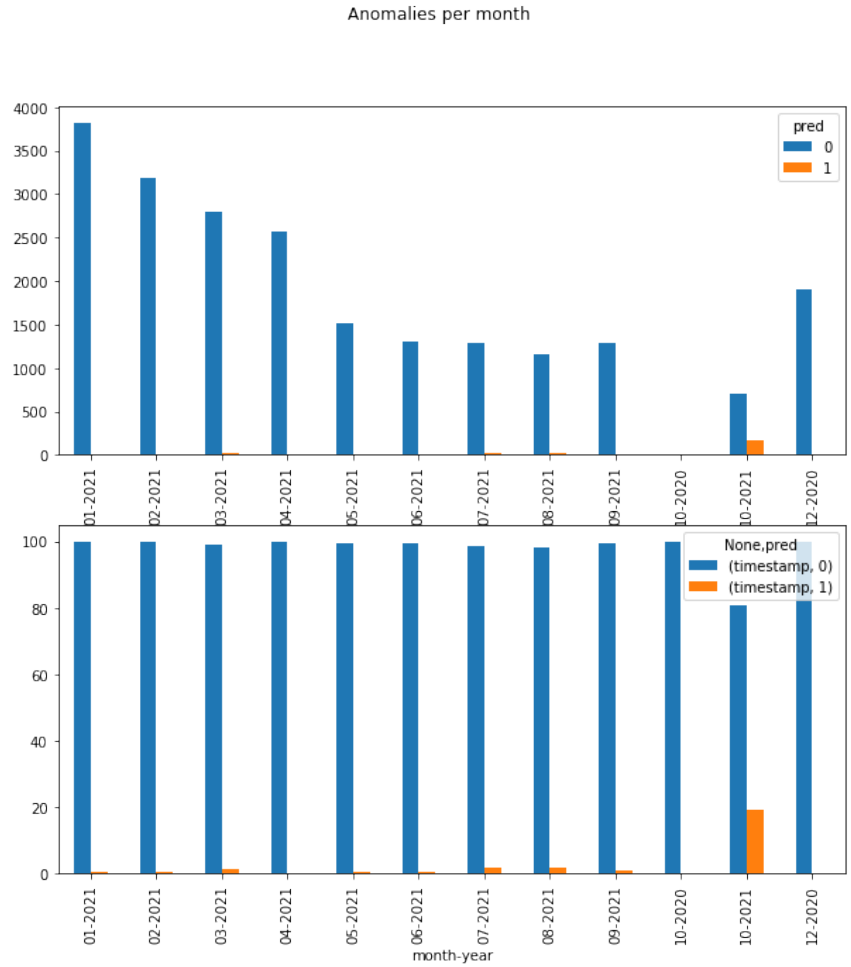

Figure 4.10: The first graph shows the number of anomalies and normal behavior per month. The second graph shows the percentage of anomalies and normal behavior per month

In Figure 4.10 can be seen the number and percentage of anomalies per month. In October of 2021, there was a high percentage of anomalies because some noisy instances were found in this period.

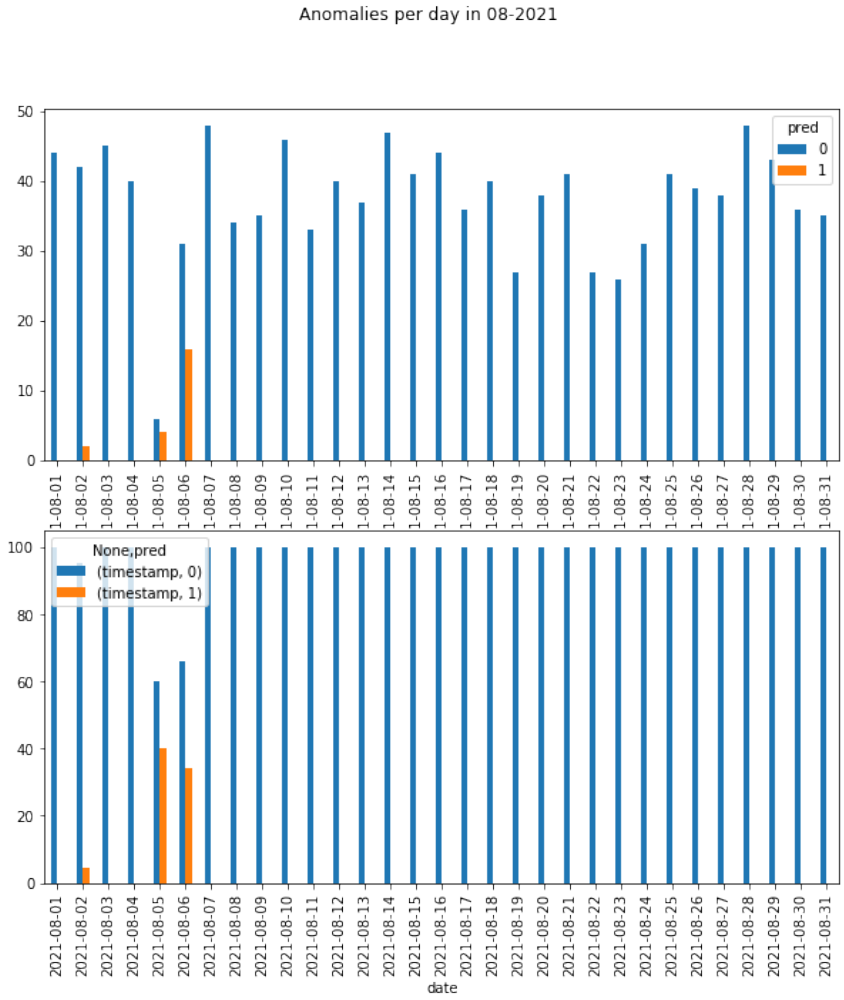

Figure 4.11: The first graph shows the number of anomalies and normal behavior in August. The second graph shows the percentage of anomalies and normal behavior in August

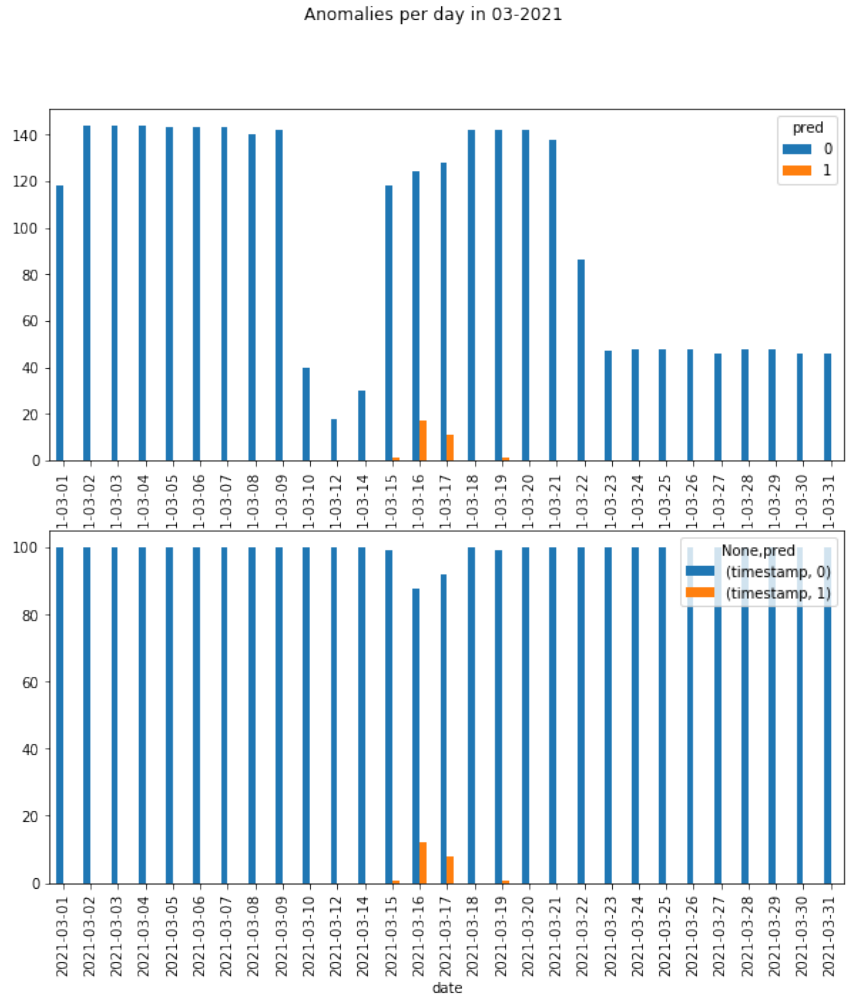

Figure 4.12: The first graph shows the number of anomalies and normal behavior in March. The second graph shows the percentage of anomalies and normal behavior in March

As can be seen in Figure 4.11 and Figure 4.12, the percentage of anomalies is higher in the periods expected to found anomalies, between 10 and 17 of March and 5 and 6 of August.

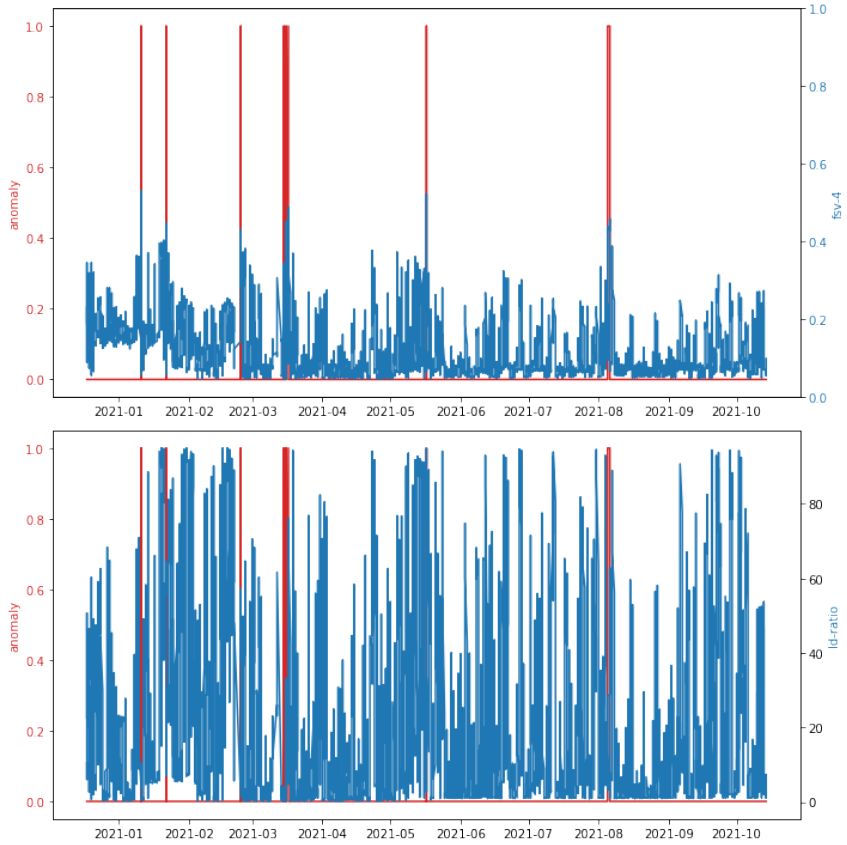

Figure 4.13: The first graph shows the occurrences of anomalies and values of fsv-4 over the year. The second graph shows the occurences of anomalies and values of ld-ratio over the year.

Finally, in Figure 4.13 it is possible to see the anomalies found over the year. It is possible to observe that the fsv-4 has a significant impact on the decision of the predictions.

4.3.1 Second phase conclusions

After receiving the new labeled data from the first phase, it was essential to perform some techniques to overcome the imbalanced data problem. The strategies used were SMOTE, TomekLink, and RandomUnderSampler.

It was performed each of these techniques with a different supervised learning algorithm to find the one that gives the best ROC AUC. The one that performed better was using Random Forest with SMOTE with 0.9526.

After analyzing the prediction for the validation dataset, it is possible to see that the anomalies were found in the expected timezone.

Chapter 5

Conclusion and future work

This final chapter includes the overview of this dissertation, the discussion of the obtained results, and the future improvements to this work.

5.1 Conclusion

This dissertation had the primary goal of discovering a way to find anomalies in datasets from electric transformers applying machine learning techniques.

In order to do that, first, the research was done and explored the literature available that was related to the subject of this thesis. This research was addressed in Chapter 2, where it was discussed machine learning techniques and methods used in predicting malfunctions in power transformers.

In Chapter 3 and Chapter 4 it was presented the development and results of the project, respectively. The project was divided into two phases: the clustering and classification phases. Initially, the goal was to find the clusters where the data could be divided to create labels. After this, the classification algorithms pass the labeled data, making the validation dataset predictions.

With the results in hand, we sent them to the company to get feedback in order to improve the model. After receiving feedback, some adjustments were made, and the model was enhanced in the best direction.

After we sent the new and improved predictions, the company discovered that these anomalies that we were trying to predict were, in fact, measurement errors on the part from the company's client. However, the company validated the project since the produced model could detect these measurement errors, bringing value to the project.

5.2 Future work

In terms of future work, there is always room for improvement in a machine learning project. Some techniques can be explored for this project as long as some improvements can be made.

One technique that can be explored is to treat this problem as a Time Series problem. This can be done once we have sequential data with timestamps.

Another noticeable improvement that can be done is, deploying the model to a machine to be used in the company.

The testing in another dataset from a different machine is also future work that can be done to test the robustness of the model.

An essential improvement is to find other types of anomalies. In this case, it is crucial that the model predicts an anomaly and indicates what features were at the cause of that decision.

References

[1] Haider Allamy. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). 12 2014.

[2] Daniel Bashir, George D. Montanez, Sonia Sehra, Pedro Sandoval Segura, and Julius Lauw. An information-theoretic perspective on overfitting and underfitting, 2020.

[3] Michael Berry and Azlinah Mohamed. *Supervised and Unsupervised Learning for Data Science*. 01 2020.

[4] Riccardo Bonetto and Vincent Latzko. Chapter 8 - machine learning. In Frank H.P. Fitzek, Fabrizio Granelli, and Patrick Seeling, editors, *Computing in Communication Networks*, pages 135–167. Academic Press, 2020.

[5] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, 2005.

[6] Haroldo de Faria, João Gabriel Spir Costa, and Jose Luis Mejia Olivas. A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis. *Renewable and Sustainable Energy Reviews*, 46:201–209, 2015.

[7] Ke Deng, Weihong Xiong, Liming Zhu, Hongzhi Zhang, and Zhengtian Li. Prediction of dissolved gas in power transformer oil based on random forests algorithm. In *2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, pages 1531–1534, 2015.

[8] Luis Guimaraes José Borge" Eduardo e Oliveira1, Vera L. Migueis. Power transformer failure prediction: Classification in imbalanced time series. 3(2):34–48, 2017.

[9] Samuel Eke, Thomas Aka-Ngnui, Guy Clerc, and Issouf Fofana. Characterization of the operating periods of a power transformer by clustering the dissolved gas data. In *2017 IEEE 11th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*, pages 298–303, 2017.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[11] Julia Handl, Joshua Knowles, and Douglas B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 08 2005.

[12] H. M. Hashemian and Wendell C. Bean. State-of-the-art predictive maintenance techniques*. *IEEE Transactions on Instrumentation and Measurement*, 60(10):3480–3492, 2011.

[13] Mohammad Hossin and Sulaiman M.N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5:01–11, 03 2015.

[14] KDNuggets. What main methodology are you using for your analytics, data mining, or data science projects? poll. Accessed: 09.01.2021.

[15] S.B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica (Ljubljana)*, 31(3):249–268, 2007. cited By 1275.

[16] Jaspreet Maan and Sanjeev Singh. Transformer failure analysis:reasons and methods. 02 2020.

[17] Farhad Maleki, Katie Ovens, Keyhan Najafian, Behzad Forghani, Caroline Reinhold, and Reza Forghani. Overview of machine learning part 1: Fundamentals and classic approaches. *Neuroimaging Clinics of North America*, 30(4):e17–e32, 2020. Machine Learning and Other Artificial Intelligence Applications.

[18] Iñigo Martinez, Elisabeth Viles, and Igor G. Olaizola. Data science methodologies: Current challenges and future approaches. *Big Data Research*, 24:100183, 2021.

[19] Piotr Mirowski and Yann LeCun. Statistical machine learning and dissolved gas analysis: A review. *IEEE Transactions on Power Delivery*, 27(4):1791–1799, 2012.

[20] Aditya Mishra. Metrics to evaluate your machine learning algorithm. Accessed: 09.01.2021.

[21] Jefferson Morais, Yomara Pires, Claudomir Cardoso, and Aldebaro Klautau. *An Overview of Data Mining Techniques Applied to Power Systems*. 01 2009.

[22] Raji Murugan and Raju Ramasamy. Failure analysis of power transformer for effective maintenance planning in electric utilities. *Engineering Failure Analysis*, 55:182–192, 2015.

[23] Vladimir Nasteski. An overview of the supervised machine learning methods. *HORIZONS.B*, 4:51–62, 12 2017.

[24] Julio-Omar Palacio-Niño and Fernando Berzal. Evaluation metrics for unsupervised learning algorithms. *CoRR*, abs/1905.05667, 2019.

[25] Alireza Sarveniazi. An actual survey of dimensionality reduction. *American Journal of Computational Mathematics*, 04:55–72, 01 2014.

[26] Shrutika S. Sawant and Manoharan Prabukumar. A review on graph-based semi-supervised learning methods for hyperspectral image classification. *The Egyptian Journal of Remote Sensing and Space Science*, 23(2):243–248, 2020.

[27] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[28] Christoph Schröer, Felix Kruse, and Jorge Marx Gómez. A systematic literature review on applying crisp-dm process model. *Procedia Computer Science*, 181:526–534, 2021. CENTERIS 2020 - International Conference on ENTERprise Information Systems / ProjMAN 2020 - International Conference on Project MANagement / HCist 2020 - International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020.

[29] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. 42(3), jul 2017.

[30] Elite Data Science. Overfitting in machine learning: What it is and how to prevent it. Accessed: 08.01.2021.

[31] Sonia Sehra, David Flores, and George D. Montanez. Undecidability of underfitting in learning algorithms, 2021.

[32] Colin Shearer. The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22, 2000.

[33] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma. A review of supervised machine learning algorithms. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1310–1315, 2016.

[34] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma. A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1310–1315, 2016.

[35] Abdulhamit Subasi. Chapter 3 - machine learning techniques. In Abdulhamit Subasi, editor, *Practical Machine Learning for Data Analysis Using Python*, pages 91–202. Academic Press, 2020.

[36] Huo-Ching Sun, Yann-Chang Huang, and Chao-Ming Huang. Fault diagnosis of power transformers using computational intelligence: A review. *Energy Procedia*, 14:1226–1231, 2012. 2011 2nd International Conference on Advances in Energy Engineering (ICAEE).

[37] Andreas Theissler, Judith Pérez-Velázquez, Marcel Kettelgerdes, and Gordon Elger. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability Engineering System Safety*, 215:107864, 2021.

[38] Eckart Uhlmann, Rodrigo Pastl Pontes, Claudio Geisert, and Eckhard Hohwieler. Cluster identification of sensor data for predictive maintenance in a selective laser melting machine tool. *Procedia Manufacturing*, 24:60–65, 2018. 4th International Conference on System-Integrated Intelligence: Intelligent, Flexible and Connected Systems in Products and Production.

[39] Jesper E. van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, Feb 2020.

[40] Sofia Visa and Anca Ralescu. Issues in mining imbalanced data sets - a review paper. *Proc. 16th Midwest Artificial Intelligence and Cognitive Science Conference*, 01 2005.

[41] Marc Wegmann, Domenique Zipperling, Jonas Hillenbrand, and Jürgen Fleischer. A review of systematic selection of clustering algorithms and their evaluation. *CoRR*, abs/2106.12792, 2021.

[42] Dennis J. Wilkins. The bathtub curve and product failure behavior part one - the bathtub curve, infant mortality and burn-in. Accessed: 10.01.2021.

[43] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168:022022, 02 2019.

[44] Lian Yu and Nengfeng Zhou. Survey of imbalanced data methodologies, 2021.

[45] Yang Zhang, Hongcai Chen, Ya Ping Du, Min Chen, Jie Liang, Jianhong Li, Xiqing Fan, and Xin Yao. Power transformer fault diagnosis considering data imbalance and data set fusion. *High Voltage*, 6, 12 2020.

[46] Lin Zhao, SH Goh, YH Chan, BL Yeoh, Hao Hu, MH Thor, Alan Tan, and Jeffrey Lam. Prediction of electrical and physical failure analysis success using artificial neural networks. In *2018 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pages 1–5, 2018.

[47] Xiaojin Zhu. Semi-supervised learning literature survey. *Comput Sci, University of Wisconsin-Madison*, 2, 07 2008.