Lisbon School of Economics & Management
Universidade de Lisboa

MASTER IN
DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK
INTERNSHIP

SEMI-SUPERVISED ACTIVE LEARNING ANOMALY DETECTION

HENRIQUE SERAFIM SANTOS

FEBRUARY - 2022

Lisbon School of Economics & Management
Universidade de Lisboa

MASTER IN
DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK
INTERNSHIP

SEMI-SUPERVISED ACTIVE LEARNING ANOMALY DETECTION

HENRIQUE SERAFIM SANTOS

ADVISOR:

FABIANA CLEMENTE
JOÃO AFONSO BASTOS

FEBRUARY - 2022

GLOSSARY

AL Active Learning. i, 3, 8, 23, 24

CV Cross-Validation. i, 14, 22

DTW Dynamical Time Warping. i, 3, 5, 6, 16

NN Nearest Neighbor. i, 18, 23, 26, 28

NRL Non-Recursive Labelling. i, 29, 30

QS Query Stategy. i, 3, 8, 24, 25, 29, 31–36

RFC Random Forest Classifier. i, 18, 21, 23, 26–29, 34, 35

RL Recursive Labelling. i, 29, 30

SC Stopping Criteria. i, 7, 8, 25–27, 31, 35, 36

SFS Sequential Forward Selection. i, 22, 27, 29

SSAL Semi Supervised Active Learning. i, 2–36, 38–40

SSL Semi Supervised Learning. i, 3, 7, 8, 23, 29, 32

ST Self-Training. i, 7, 8, 23, 24, 31, 34, 36

SVM Support Vector Machine. i, 7, 23, 24, 26, 27

TS Time Series. i, 1–17, 19, 20, 26, 27, 34–36, 41

TSFEL Time Series Feature Extraction Library. i, 21, 26–29, 36

ABSTRACT

The analysis of Time Series data is a growing field of study due to the increase in the rate of data collection from the most varied sensors that lead to an overload of information to be analysed in order to obtain the most accurate conclusions possible. Hence, due to the high volume of data without labels, automatized detection and labelling of anomalies in Time Series data is an active area of research, as it becomes impossible to manually identify abnormal behavior in Time Series because of the high time and monetary costs.

This research focus on the investigation of the power of a Semi Supervised Active Learning algorithm to identify outlier-type anomalies in univariate Time Series. To maximize the performance of the algorithm, we start by proposing an initial pool of features from which the ones with best classification power are selected to develop the algorithm. Regarding the Semi Supervised Learning segment of the process a comparison between several classifiers is made. In addition, various Query Strategies are proposed in the Active Learning segment to increase the informativeness of the observations chosen to be manually labelled so that the time spent labelling anomalies could be decreased without a great impact in the performance of the model.

In a first instance, we demonstrate that the pool of designed features better identifies the anomalies than features selected in a fully automatized process. Furthermore, we demonstrate that a Query Strategy used to select the most informative observations to be expertly classified based on the utility and uncertainty of the observations exhibit better results than randomly selecting the observations to be tagged, improving the performance of the model without infeasible time and cost spent in the identification of the anomalous behavior.

KEYWORDS: Time Series Anomaly Detection; Feature Extraction & Selection; Semi Supervised Active Learning; Query Strategy Informativeness

TABLE OF CONTENTS

List of Figures

List of Tables

ACKNOWLEDGEMENTS

Firstly, I would like to thank my Ydata supervisor, Fabiana Clemente, for the guidance, significant contributions and for all the time spent clarifying my doubts and concerns.

Secondly, I would also like to thank to my ISEG supervisor, Professor João Afonso Bastos, for all the valuable tips.

I am also grateful to the whole Ydata team for receiving me with open arms and always be available to discuss any problem.

Finally, I would like to thank my family for supporting me on this journey.

SEMI SUPERVISED ACTIVE LEARNING ANOMALY DETECTION

By Henrique Serafim Santos

The goal of this research is to develop a Semi Supervised Active Learning algorithm capable of identifying outlier-type anomalies in univariate series. To accomplish this, we engineer a set of features and compare the performance of several classifiers and query strategies so that the performance of the model is maximized in situations of label scarcity.

1 INTRODUCTION

Time Series (TS), ordered sets of data points indexed over time, have been an important research field for a long time as they enable the study of real-world systems' dynamics. The growing interest in this form of data led to an increase of the importance of the variable 'time', as its introduction allow the data analysts to obtain insights that are not possible to reach in its absence. Thus, as the component 'time' is present in almost every action or measurement, it was only logical to introduce this component in the analysis made and it can even be said that TS have become a ubiquitous form of data. In this thesis, we will focus our study on univariate TS, i.e., TS with only one variable besides the temporal one and that can be formally defined as

$$\dot{T} = (t_0^d, t_1^d, ..., t_n^d), t \in \mathbb{R}, d = 1, n \in \mathbb{N} \tag{1}$$

with n being the length of the TS and, generally, with the observations evenly space. Naturally, TS anomaly detection has become an active and dynamical area of research as, most of the times, it makes no sense to conclude that a single point in time and the measures obtained at that moment are abnormal, ignoring the temporal context in which those measures were collected. Then, the detection of anomalies on TS can be described as the attempt to predict whether a newly observed TS behaves normally or abnormally when compared to the TS behavior previously observed. This field of study is central for numerous domain applications, such as finance, medicine, or business, allowing to identify fraudulent transactions (Lorenz et al., 2020), to diagnose heartbeat patterns associated with a decease (Chuah and Fu, 2007) and to detect production irregularities (Liang et al., 2021), respectively. Furthermore, it is also very useful in human activity recognition (Machado, 2013), video surveillance (Au et al., 2006), signal recognition (Teng, 2010) and other monitoring applications. The methodology behind anomaly detection is specific of the type of data, being the methods used for detecting anomalies in images different from the ones used for the detection of anomalies in data streams. What is common

across all types of data is that, usually, the data analysts are more interested in the anomalous events (like heart deceases or fraudulent transaction) than in the normal behavior and so it is preferable to identify a normal sample as abnormal than the opposite. However, the training dataset, i.e., the observed data, is normally imbalanced, having many more samples classified as normal than as abnormal, making it harder to differentiate these two types of data and posing one of the main challenges of the anomaly detection problems. As more and more devices capture temporally dependent data, we witness an exponential increase in the volume of this kind of information available for analysis, elevating the relevance of this topic to higher levels and making it critical in the current paradigm. This already huge and continuously growing volume of data makes it impracticable to label the anomalies manually, as the time and cost spent on that activity are unbearable for most of the companies.

To tackle this problem, several automatized techniques were developed. Initially, to avoid the manual labelling of all new observations, and to allow an automatic classification of some new samples without the necessity of an expert, statistical approaches were proposed to detect the anomalies present in TS (Tukey et al., 1977; Chang et al., 1988). Generally, these methods admit that the data are generated by a specific statistical model. However, with the continuously growing computational power in recent decades, researchers introduced classical Machine Learning algorithms as a viable way for detecting abnormal behavior in TS. More recently, the great progress achieved by Deep Learning approaches in computer vision tasks motivated the use of models, such as Multi-Layer Perceptrons, Convolutional Neural Networks and Autoencoders, to anomaly detection. In these approaches, as long as the algorithm is able to return accurate results, the underlying data generation process is not considered relevant, being these processes usually black-box methods simply trying to learn from the observed data. However, most of these algorithms fall into one of the following two categories:

- Assume the access to the labels of almost all observations or at least of a considerable quantity of them
- Are computationally too expensive

In the real-world, most of the times, only a small number of labelled data is available, opposing to the large quantity of unlabelled data, which invalidates the use of the algorithms that felt in the first category. Due to this scarcity of labels and to the complexity of the methods that can perform without them, there is an increasing need to secure the applicability of different efficient methods to perform anomaly detection.

To increase the performance in the classification of anomalies with scarce quantity

of labels, it is possible to use the power of the copious amounts of unlabelled data that are often available. Thus, addressing anomaly detection through Active Learning (AL) emerged as a logical path to follow. The goal is to decrease the quantity of manually assigned labels needed without compromising the performance of the anomaly detection algorithm. Hence, an algorithm able to choose the best samples to label among the ones that are unlabelled should be implemented. To achieve it, in this paper we will combine the power of Semi Supervised Learning (SSL) and AL by developing a Semi Supervised Active Learning (SSAL) algorithm for anomaly detection in univariate TS with observations evenly spaced. The model should be able to automatically attribute labels to TS slices and, when the certainty in the attribution of the labels starts to decrease, be able to select the most informative unlabelled observations to be labelled by an expert. As a consequence, there is a necessity of establishing a similarity measure between pairs of TS. Over time, various approaches were suggested, from the widely used Euclidean distance and its derivatives to more complex methods such as Dynamical Time Warping (DTW) (Berndt and Clifford, 1994; Müller, 2007) but all of them have their own limitations. Moreover, unlike in anomaly detection on non-temporal data, in which the data points are independent from each other, we need to assume that the observations are not completely independent, having the latest observed values influence over the following data points. The conjugation of these factors led researchers to propose that the measurement of similarity between TS should be done by extracting some features capable of summarizing both the characteristics and the relation between consecutive timestamps. (Fulcher and Jones, 2014; Fulcher, 2017)

Having said that, in this research we propose an initial set of features and a selection process able to elect the ones that best capture the TS characteristics that allow to differentiate normal from abnormal behavior. We compare the performance of different classifiers to suppress the bias in the selection of the features. Then, we compare the performance of different Query Stategy (QS) by introducing them in the conceived SSAL model. Finally, we study the trade-off between model's performance and cost.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the dataset used to develop and evaluate the performance of the model developed. Section 4 presents the methodology proposed in this thesis, including the feature extraction and selection processes and the algorithm used to perform the anomaly detection. In Section 5, the performance of the model is evaluated and a comparison between the different analysed methods is done. Finally, in section 6, the main conclusions are discussed.

2 LITERATURE REVIEW

2.1 Window Size

When performing anomaly detection in TS, it is necessary to frame the observations since by observing an isolated point we do not have the context needed to classify it. Thus, it is imperative to find a solution to divide the data in such a way that the best possible framing can be achieved, leading to the extraction of more informative data. Having access to the labels, the most common method used is to simply attribute the label observed to stretches of consecutive timestamps with the same label. This is in line with Machado (2013), where it is referred that if the subject is performing a single activity, a longer window will include more information about it. However, this approach is neither valid for TS for which we do not have access to the labels nor for many real-world datasets, in which the assumption of the existence of long stretches of timestamps with the same label does not hold. As an alternative, the two more explored types of window segmentation are *static window*, in which the division is performed by consecutive windows without gaps, and *sliding window*, in which a percentage of samples are overlapped between consecutive windows (Bota, 2018), i.e., a single sample can belong to two or more different windows and prevent the cut of the signal in inconvenient locations (Aggarwal, 2005). In both cases, the TS are divided into equally sized windows. The choice of the number/length of frames should take into account the trade-off between the quality of information obtained and the resolution, as smaller windows would increase the size of the training set at the cost of higher computational complexity. What seems to be consensual across previous literature is that there is not an optimal window size across all datasets and so it should be considered as a tuning hyperparameter.

2.2 Similarity between Time Series

Besides the choice of the window size to enable the slicing of large TS into smaller stretches, the way to measure the similarity between these slices is of extreme importance. The implementation of a similarity measure allows the comparison of the behavior of a certain TS slice with normal and anomalous behaviors previously observed, validating the classification of this with the label of the most identical one.

The widely spread Euclidean distance and other metrics commonly used to obtain the distance between two samples for spatial data such as the Manhattan, the Chebyshev and the Minkowski distances were extrapolated to the comparison of TS of the same length and for which the observations are equally spaced. Nevertheless, these measures do not perform well when the two TS are in different phases and are not even applicable if they

have different length. This happens because these metrics are only able to compare the value of both TS at the same timestamp.

Keogh and Batista (2013) demonstrated the poor performance of the Euclidean distance in many TS classification cases and to overcome it, proposed a DTW approach. This alternative, explored in several papers (Berndt and Clifford, 1994; Müller, 2007), consists of finding the optimal alignment between two TS by stretching or shrinking one of the TS to match the other. This permits the two compared temporal sequences to have different speed and/or length and to be in different phases. The objective is to find the minimal distance warp path by dividing the problem into sub-problems and saving the solutions of those on a cost matrix. Unfortunately, DTW is computationally expensive, scaling quadratically relatively to the TS length and limiting its use to small TS datasets, as comparing TS containing only 177000 measurements requires memory in the terabyte range (Salvador and Chan, 2007).

To speed up the DTW, some constraints limiting the number of evaluated cells of the cost matrix while finding the minimal distance warp path were imposed, namely the Sakoe-Chiba Band (Sakoe and Chiba, 1978) and the Itakura Parallelogram (Itakura, 1975). These approaches perform well when the optimal warp path is expected to be similar to a straight line crossing the diagonal of the cost matrix. However, these constraints only speed up DTW by a constant factor, being the algorithm still $O(N^2)$. An alternative is Data Abstraction, that speeds up DTW by running it on a reduced representation of the data (Chu et al., 2002; Keogh and Pazzani, 2000), finding a warp path for the lower resolution TS and mapping it back to the full resolution cost matrix. Nevertheless, although speeding the algorithm up by a large constant, it is once again still $O(N^2)$.

Salvador and Chan (2007) proposed Fast DTW, that uses a multilevel approach and can find an accurate warp path in linear time and space by combining ideas from the previously presented alternatives. With Fast DTW, although the time to calculate the similarity between two TS is almost negligible ($\approx$ 0.01 seconds for TS of length 100), the same cannot be said of the computation time for all the distances, as this approach still scales quadratically with the number of TS. For example, for a small dataset of 10000 TS, it would take (10000*9999/2) * 0.01 $\approx$ 139 hours to calculate all similarities.

Then, assuming that we want to calculate the distance between all TS, the solution must pass through a reduction in the time spent in the calculation of the distance between each pair of TS. With that goal in mind, an approach involving obtaining the similarity of TS based on extracted features was proposed (Fulcher and Jones, 2014; Fulcher, 2017) and allowed to acquire an interpretable summary of the dynamical characteristics of each TS and save it in a vector. The similarity between the obtained feature vectors can then

easily be measured with simple metrics as the Euclidean distance. Thus, although this methodology still scales quadratically with the number of TS, this is attenuated by the fact that the calculation of the similarity between two feature vectors using Euclidean distance is incredibly faster than using Fast DTW. In the next subsection, we will visit some of the alternatives proposed using this approach.

2.3 Feature Extraction

Feature extraction is a process of dimensionality reduction by which the raw data are reduced in such a way that fewer features can capture the same information. Usually, this approach yields better results than when the raw data are directly used. Besides, feature-based representations do not demand TS to have the same length. With the goal of being able to identify anomalies as accurately as possible, it should be selected an appropriate feature-based representation of the TS, i.e., a feature vector composed by features with high discriminating ability. To accomplish this, the features to be extracted could be chosen solely based on the expert analysis of the dataset, without quantitative comparison across different sets. To avoid the bias introduced by that methodology, as it is uncertain whether other set may have had a better performance, the feature vector can be obtained through systematic comparison across a comprehensive TS feature library. However, this is computationally expensive, avoiding the dissemination of libraries of feature-based representations of TS for real world applications. Lubba et al. (2019) proposed a minimally redundant set of 22 features that exhibited great performance across a set of 93 TS classification datasets (containing over 147000 TS) of different fields of study. This set of features, known as catch22 (CAnonical Time-series CHaracteristics), was selected by performing dimensionality reduction from a set of 4791 features from the hctsa toolbox and it computes quickly ($\approx$ 0.5 s/ 10,000 samples, roughly a thousand times faster than the full hctsa feature set in Matlab), despite an average reduction in classification accuracy of just 7%. However, Bastos and Caiado (2021) stated that in certain domains, namely the financial, «well-informed expert knowledge may not be disregard in favor of agnostic representations of the data». Machado (2013) presented a set of 18 features for Human Activity Recognition, belonging to 3 domains: statistical, temporal, and spectral. For the same field of study, Bota (2018) proposed a renewed set of 30 features, with 13 features in common with the previous paper proving that this is a dynamical area of research and that there is not an absolute answer to which is the best set of features.

2.4 Model

Over the years, various supervised approaches to the problem of classifying TS as anomalous or normal have been proposed. Braei and Wagner (2020) conduct a survey of

these methods. Statistical methods, like the ARIMA and its derivatives or the Simple Exponential Smoothing are compared against classical Machine Learning approaches such as the One-Class Support Vector Machine (SVM), the Local Outlier Factor or the Extreme Gradient Boosting (XGBoost). Representing Deep Learning alternatives were put to test Multi-Layer Perceptrons, Convolutional Neural Newtworks, Long-Short Term Memory Networks, Autoencoders among others. It was concluded that, despite the advances in Machine Learning and deep neural networks, statistical methods generally perform better and have the extra advantage of being faster and easily interpretable. However, this performance is accomplished at the cost of having a large amount of labelled data. In general, the labelling of ground truth data must be done manually, being prohibitively costly both in terms of time and money. When there is a scarcity of labels, it is possible to take advantage of the copious amount of unlabelled data available to improve the performance of the model. Despite the idea might seem unintuitive, different studies demonstrated the utility of this procedure. (Cohen et al., 2004; Ganesalingam and McLachlan, 1978).

Considering this, one of the methods proposed to perform anomaly detection in TS when the quantity of labels is scarce was SSL, which is a technique able to automate part of the labelling process, while reducing the labelling cost and the need for great volumes of labelled data a priori. Some SSL methods, such as the proposed in Wei and Keogh (2006) automatically and iteratively classify as anomalies the unlabelled samples with higher similarity to the already labelled abnormal samples. However, it is not easy to decide when to stop this iterative method in such a way that the classification accuracy does not start to decrease due to systematic misclassification of normal observations. To tackle this problem, they proposed a heuristic for the Stopping Criteria (SC) based on the change of the minimal nearest neighbor distance between labelled samples, claiming that when normal samples start to be misclassified, this distance tends to decrease due to the higher density regions in which normal samples are inserted. Nguyen et al. (2011) presented a technique named Learning from Common Local Clusters in which it is performed the clustering of the unlabelled samples and all samples within a cluster are assumed to be from the same class, but this assumption is unpractical. Wang et al. (2019) proposed a SSL model based on shapelets, that are maximally discriminative features, in which the unlabelled samples are treated in supervised fashion by using pseudo-labels. However, this process of learning can be extremely inefficient because TS normally have an enormous number of candidate segments. There are also SSL methods that perform finely when the dataset considered is completely labelled but only consists of normal points (Braei and Wagner, 2020). Wei and Keogh (2006) state that SSL methods can be organized into five classes: SSL with generative models, SSL with low density separation, graph-based methods, co-training methods, and Self-Training (ST) methods. **Generative methods**

admit that data are obtained from a mixture distribution that can be identified by large amounts of unlabelled data. Hence, knowledge relative to the structure of the data can be incorporated in the model. **Low density separation methods** exploit the supposition that «the decision boundary should lie in a low density region». However, Keogh et al. (2005) demonstrated that «(abnormal time series) do not necessarily live in sparse areas of n-dimensional space» and «repeated patterns do not necessarily live in dense parts». **Graph-based approaches** assume that «the (high-dimensional) data lie (roughly) on a low-dimensional manifold» and represent the data as nodes and the distances as edges. However, the graph must be manually constructed for each domain. **Co-training**, firstly proposed in Blum and Mitchell (1998), assumes that the features are independent, separating the features of the data into two disjoint sets, training two classifiers and using the predictions of one of the classifiers to enlarge the training set of the other. Notwithstanding, TS normally present very high feature correlation (Keogh and Kasetty, 2003), disabling the use of this method. One of the simpler and better performing methods is **ST**, in which a classifier starts by being trained with a tiny quantity of labelled data and then labels the unlabelled samples for which it has a higher certainty. The procedure is repeated, being the classifier trained with a bigger training set at each iteration.

An alternative to SSL is AL, which is a method composed by the QS, that selects the most valuable instances to be labelled and incorporated into the classifier's training set, and the Oracle, that annotates the selected samples. The procedure is replicated until the addition of new labelled samples to the training set does not increase the classifiers accuracy or until some SC is met. The main idea behind AL is to use the user expertise to enlarge the training set and enhance the confidence of the remaining predicted labels. He et al. (2015) showed that a «smaller number of training data may learn a better learning model for classification when it consists of more informative examples». Thus, the sample selected to be manually labelled should be the one for which the classifier is more uncertain about the label attribution. However, experimental results demonstrated that, sometimes, an uncertainty-based QS tends to select outlier samples rather than boundary ones (Zhu et al., 2009), and the addition of these to the training set would not increase the discriminative power of the classifier and would, ultimately, introduce bias to the classifier. Hence, to overcome this problem, it was utilized a sampling strategy that combined both the uncertainty of the sample and the local data density of it (He et al., 2015, 2017; Zhu et al., 2008). This way, the selection is made both in terms of uncertainty and representativeness, increasing the value of the selected sample to the classifier's training set. Settles (2009) contains a comprehensive survey of different proposals in the field.

To leverage both approaches, He et al. (2015) developed a SSAL that, after every new annotation done by the Oracle, runs a SSL algorithm to rapidly augment the training set.

3 DATA

In this study, it will be used the *Yahoo! Synthetic and real time-series with labelled anomalies, version 1.0* dataset to develop and calibrate a model able to perform anomaly detection on TS data. Part of the same dataset will then be used to evaluate the results obtained. The dataset is made available[1] as part of the *Yahoo! Webscope program*, to those who have signed a Data Sharing Agreement and agreed to use it for approved non-commercial research purposes, precluding the option of redistributing it.

It consists of an assembly of real and synthetic TS, with anomalous points properly identified, divided into 4 different sets of files in which each file corresponds to one TS. Thus, the first set is composed by 67 TS representing several Yahoo production traffic property metrics. The remaining sets have 100 TS of synthetic data each, created with varying trend, noise, cyclicity and/or seasonality. Each file is composed by 3 columns, representing the timestamp, the value measured or synthetically created and the label.

The *timestamp* is presented as a range from 0 to the length of the TS in the real set and as a UNIX timestamp for the synthetic data, representing each data point 1 hour worth of data in both cases. Furthermore, the synthetic TS have fixed length depending on the set to which they belong, with the TS from the second set presenting 1421 hours of data and the ones from both third and fourth sets having 1680 data points each. On the other hand, the number of data points of the set with real TS is variable, which is easily justifiable by the fact that different TS represent different Yahoo! services, ranging from 741 hours to 1461 hours but mainly concentrated between 1420 and 1461 hours (65 of the 67 TS are in this situation). Taking into account all observations, the dataset presents a total of 572 966 hours of annotated behavior.

The variable that records the measures obtained across time is named *value* and it is represented by a numeric that can take a wide range of values, both positive and negative. In Figure 1, we can observe the distribution of this variable and conclude that the dataset has TS moving around median points of different orders of magnitude. The figure also demonstrates that the absolute value of the variation between consecutive data points vary across different magnitude orders, originating a diverse dataset in which we may have TS taking values like

(..., 6933764, 6998573, 6954333, 6413081, ...), (..., 32, 23, 24, 39, ...),

(..., -495.61, -718.19, -540.78, -749.97, ...), (..., 0.0878, 0.1154, 0.0734, 0.0404, ...),

(..., 39.96, 38.68, 39.43, 39.48, ...), (..., 12715, 12736, 12716, 12739, ...).

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

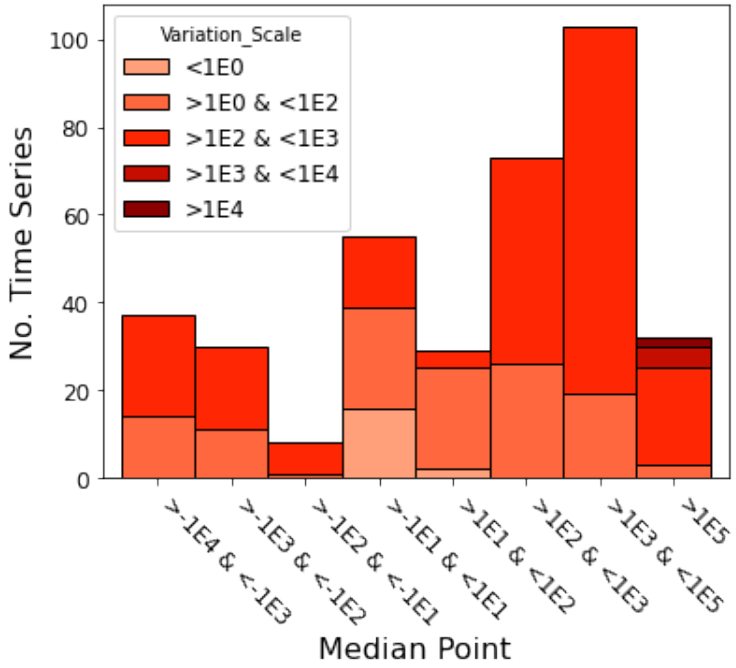

FIGURE 1: Time Series Median Point and Variation - Orders of Magnitude

The dependent variable, named *is_anomaly* in the first two sets and *anomaly* in the other two, takes value 0 if the observed measure of *value* is considered to behave normally and takes value 1 if there is reason to believe that the observed value is abnormal when compared to the remaining behavior of the sequence. In the pool of all observations there are 3915 data points labelled as anomalies, which represents only 0.68% of all observations. Moreover, the consecutive number of data points tagged as anomalies is not constant, varying from an isolated abnormal point to a maximum of 114 consecutive anomalous data points. Considering as an anomaly a stretch of consecutive abnormal data points, Table I shows the representation of each type of anomaly.

TABLE I: LENGTH OF ANOMALIES

	Number of Anomalies	% Total Anomalies	Cumulative %
1	1869	86.85	86.85
$2-5$	233	10.83	97.68
$6-24$	36	1.67	99.35
$25-100$	11	0.51	99.86
>100	3	0.14	100

Isolated anomalous data points are a big portion of the total number of anomalies tagged. This was expected as, after a first view of the available data, we could conclude that the anomalies tagged are the result of sudden variations of the *value* variable, causing spikes in the TS. These spikes are mostly a result of an atypical increase followed by

a similar amplitude decrease, but the opposite also happens. The longer anomalies are result of an initial anomalous variation that impacts the following data points, being the abnormally big variations propagated to these data points. Nevertheless, only a residual 0.65% of the anomalies take more than 1 day to return to normality. Some examples of the described anomalies can be found in Figure A.1. Nevertheless, there is an infinitesimal percentage of anomalies that are a result of unexpected stability in highly variable TS as illustrated in Figure 2

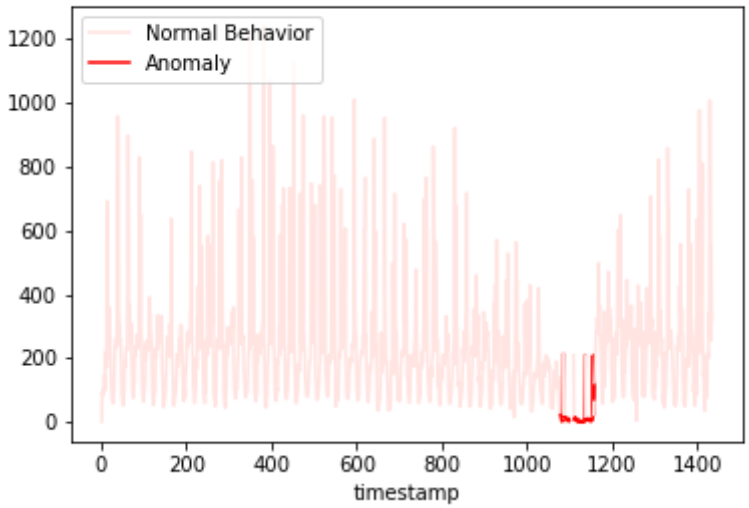

FIGURE 2: Unusual Type of Anomaly

Moreover, the abnormal behavior is not equally dispersed across the TS, being concentrated near the middle of the TS and close to the end of these ones as illustrated in Figure 3.

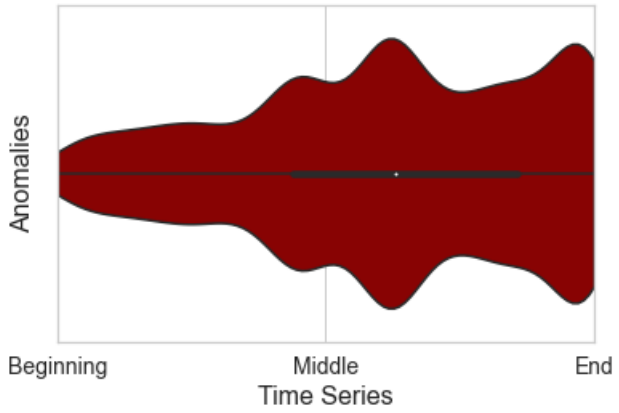

FIGURE 3: Anomaly distribution across Time Series

Regarding the synthetic datasets, the anomalies are inserted randomly and the difficulty in its identification is gradually increased, having the TS of the second set a low level of noise, facilitating the identification of the anomalies at first sight, while the anomalies from the fourth set require more attention and time to be properly classified. On the other hand, the anomalies in the real set are marked by humans and therefore may not be consistent. An example of the subjectivity in the classification of TS data points can be found in Figure A.2 in which two more or less similar data points receive different classification regarding their abnormality.

The four sets of data are equally spaced, do not have missing data, and although different TS exhibit different magnitude orders, the type of anomalies observed is similar across all TS, allowing us to create a single dataset by joining the 4 sets of data. In the next section, with this joint dataset, a model will be developed to better identify the anomalies in a scenario in which a big chunk of the labels is hidden to replicate an environment in which the labels are unknown.

4 METHODOLOGY

4.1 Preprocessing

Before developing the model, we need to separate the data into a train and a holdout set in which the results obtained will be evaluated. One popular technique is to separate a chunk of data at the end of each TS, conceding a greater representation to the holdout set. However, as shown is section 3 the anomalies are not homogeneously dispersed across the TS length, and so, the use of this technique would imply a higher representation of anomalies in this set than in the original data, which is not advisable. The alternative chosen to avoid this problem was to separate some TS based on the assumption that the distribution of anomalies across different sets of TS is more or less similar. To accomplish the separation on a random manner it was used the seed 54391[2] and approximately 10% of the original data was attributed to the holdout set that will not have any kind of influence in the model development as this set will be used as unseen data while testing the effectiveness of the refined process. Nevertheless, to ensure the balanced representation of the pre-joint datasets, the random selection of the TS was done within each initial dataset, resulting in the selection of 7 real TS and 10 TS from each one of the synthetic sets, adding up to a total of 37 TS.

After obtaining the two sets, we aim to transform the raw data so that the input passed to the model is cleaned to improve performance. To achieve this, a preprocessing pipeline needs to be created. As discussed in section 3, the TS to which the model will be applied have varying trend. Nevertheless, the type of anomalies that we intend to identify seem to be results of sudden changes on the data, not being influenced by the level in which this behavior happen. Due to that, we believe that removing the trend component will not benefit the model performance and will introduce an unnecessary bias. In the TS description, we also addressed the variability in the seasonal and cyclical components. In fact, there are some TS that presented a seasonal component that could be removed, facilitating the identification of the anomalies through the model. However, the period of this seasonality is different from one TS to another, implying the use of an algorithm able to identify the seasonal period and then perform the removal of this component. Preliminary tests were run in the developing of such algorithm. However, the existence of a high number of TS that have cyclic behavior instead of seasonal led this prototype algorithm to, for some cases, wrongly remove a seasonal component that, in fact, did not exist and so making it harder for the model to identify anomalies in TS in these conditions due to the creation of spikes that were not supposed to exist. Therefore, we believe that

[2] Author's ISEG Student Number

the time spent to remove the seasonal component is not justifiable with an increase on the model performance and so we decided not to perform the removal of this component.

Another problematic raised while analysing the dataset was the different scales of the various TS. A natural approach would be to scale the TS, by applying a Normalization or Standardization scaler. Nonetheless, we intend to use Cross-Validation (CV) in a future step and applying these methodologies before CV leads to a leak of information about the holdout data distribution into the way the training data are scaled. Furthermore, as neither the median of the variation scale nor the level in which a TS is nested influence the classification that should be attributed, we decided, once again, not to apply any transformation and deal with the problem of having different scales in a future step.

We already discussed that most of the anomalies are the result of sudden increases followed by sudden decreases or vice-versa, and so, we intend to obtain the magnitude of the differences between consecutive data points. We intend to do it before dividing the TS into stretches merely because of the lower time cost of calculating the differences in a dataframe containing all observations of a TS (and then dividing these differences along with the values into stretches) when compared to the time spent to calculate the differences in a set of arrays representing the stretches. This cost discrepancy is more evidenced when some kind of overlap is used as some differences would have to be calculated twice when following the second alternative.

Some TS present initial values (mainly in the first 3 data points) that seem like anomalies but are not classified like that. This behavior can be observed in Figure A.1 b), in which the two first data points do not fit with the remaining values of the TS. Moreover, there are some TS for which the first observation takes value 0 and that value is not repeated across the remaining data points. In both situations, we believe that these annotations may corresponds to an initialization of the sensors that are capturing the data. Thus, we decided to drop the initial values that fit in one of the two following categories:

- Take value 0 at the first data point and do not record any value less or equal to 0 between the fourth and hundredth data points
- Are further than three standard deviations from the mean of the data points, being both these measures obtained from the data points between the fourth and the last of the first window.

As discussed in subsection 2.1, one approach for TS division is based on the idea that longer windows representing the same activity are more informative about it. Following this methodology, we would be able to capture higher power explanatory windows with

regard to the normal behavior. Notwithstanding, as shown in section 3, the anomalies that we are trying to identify are predominantly isolated data points. As we cannot consider a single point a TS, we would have to expand the window a certain number of observations centered around the anomaly. However, this could only be applied to the TS from the training set, i.e., for the ones that we know the labels a priori, having the unseen TS to be divided in a different manner because of the unknown position of the anomalies. Therefore, this method has as disadvantages the difference in the size of the windows between observed anomalies, observed normal behavior and unobserved data (that could lead to the association of smaller windows to abnormal behavior) and the high degree of personalization needed in the division of the training set. Due to these complications, we decided to divide the windows by using a sliding window of fixed size that will travel across the TS, covering all the observations and tagging as anomalies the stretches that contain at least one data point classified as anomalous. This approach does not take advantage of the informative power of longer windows classified as normal and can lead to the division of the TS in inconvenient regions, complicating the process of anomaly identification. To minimize the risk of those undesirable cuts and to increase the number of samples, we decided to use an overlap of 50% and compare its performance against the absence of overlap. As window size, we decided to utilize 24 because the observations are hourly separated and that way, we guarantee windows containing one day worth of data.

By choosing a window size of 24 with 50% of overlap, the stretches at the end of some of the original TS will be shorter, as the lengths of some TS are not divisible by 12. Nevertheless, due to the high variability of the length of the TS discussed in section 3 there was not a common divisor for all TS and so this would always happen for a subset of them. To work around this issue, the stretches in this situation will be completed using a technique named masking, ensuring that all stretches have the same length, without affecting the characteristics of the stretch itself. In the absence of overlap, in addition to this, we may face a situation in which a stretch of 24 only contains one value observed, being the remaining 23 masked values. To avoid it, before performing the division of the TS for which the length divided by 24 returns a remainder of 1, we decided to drop the first data point of the TS. Although losing a small portion of information by dropping this observation, this will not negatively affect the identification of anomalies, because, as we showed in Figure 3 the data points classified as anomalies are mostly concentrated in the second half of the TS and so the probability of dropping an anomaly point is infinitesimal.

The preprocessing pipeline is summarized in Table II and it is applied after the separation of the holdout set.

TABLE II: PREPROCESSING PIPELINE

Input
set_series, window_size[(a)], overlap[(b)], seed[(c)], mode[(d)]
Process
Eliminate Initial Outliers
Create Differences
Divide Time Series
Feature Extraction from the stretches
Shuffle to Homogenize the set
Output
Ideal Feature Vector

[(a)] 24 to retain 1 day worth of data
[(b)] True or False
[(c)] 54391 to reproduce the results
[(d)] 'value' or 'diff'

The motivation behind the decision of using Feature Vectors and the process of selecting which features should be extracted will be discussed in Subsections 4.2 and 4.3

4.2 Feature Engineering

As discussed in Subsection 2.2, there are several ways to measure the similarity between TS. One of the proposed methods was the computation of the distances of TS' data points simply by using Euclidean distance. Nevertheless, with this strategy, if an anomaly occurs in the beginning of one window, the similarity with a stretch of normal behavior will be greater than the similarity with a stretch with an anomaly in the end of the window. Other strategy was the Fast DTW (Salvador and Chan, 2007) as opposed to the more widespread but slower DTW. However, in the best-case scenario, i.e., when we do not use overlap and we have a lower number of stretches, this methodology still performs exceedingly slow. In fact, knowing that we have 330 TS in the training set and that the average length of them is at least 1400, we would have as lower limit for the number of stretches 1400*330/24=19250. Hence, as the time to calculate the similarity between two TS of length 24 is approximately 0.001[3] seconds, we would have to spend (19250*19249/2) * 0.001 $\approx$ 51 hours to calculate the similarities between all slices. Besides, both these methods are scale dependent, and so two stretches with the same pattern but different scales would be considered less similar than two stretches with similar scale but completely different patterns.

Meanwhile, also in approximately 0.001 seconds, it is possibly to calculate the Euclidean distance between 2 vectors of length 300000, proving the compared efficiency

[3]Using an Intel® Core ™ i5-10400 CPU @ 2.90 GHz, 2904 Mhz,6 Nucleous,12 Logical Processors

of this approach. Thus, we intend to extract some features capable of summarizing the behavior of a slice and saving them in a vector, being the distance between vectors of different slices calculated using the Euclidean distance and used as the measure of similarity. Moreover, by following this methodology we are able to engineer features that are not scale dependent, allowing the comparison of the stretches obtained from different original TS.

Bastos and Caiado (2021) concluded that, for some domains, features created based on expert knowledge are better than agnostic representations of the data. Hence, we decided to engineer our own set of features, aiming to maximize the performance of the model.

As we are dealing with TS data, it is imperative to keep some information regarding the original TS from which the stretch was extracted. To achieve this, a pre-selection of features is extracted, including measurements about the whole original TS, the chunk of data that occurred before the stretch or even a smaller number of data points that were observed rightly before the stretch. To this set of features, along with some measures related to the stretch itself, we attributed the name 'General Feats'. The description of these features can be consulted in Table A.I. Some of these pre-extracted features include summaries of the metrics obtained across stretches instead of measurements of the original data points which introduces an unwanted bias. Nevertheless, the performance of such features has a sufficiently satisfactory performance for us to overlook that bias in an attempt to improve the performance of the developed model.

To improve the performance of the model, we developed a pool of 162 features that result from different calculations using the pre-extracted features and being many of the features in this pool slight variations from others in order to maximize the discriminant power of those. To be included in this pool of features, the candidates had to satisfy three criteria.

- Have to be scale independent, i.e., slices with different scales but similar pattern have to yield similar results, as we intend to calculate the distance between feature vectors later.
- All of them should be dispersed around similar values so that, when the computation of the distance between two feature vectors is performed using Euclidean distance, this measure is not monopolized by the distance of a single feature moving in a higher scale than the others.
- Must perform satisfactorily, i.e., yield a metric above a certain threshold, when used alone to classify the stretches, using 20% of the training set as validation set. To accomplish this, we averaged the F1-scores (the use of this metric will be

addressed later) obtained from a Random Forest Classifier (RFC) with 5 trees and a 5-Nearest Neighbor (NN) and dropped the candidates that yielded a F1-Score below 0.5. These two simple classifiers were used to allow a fast and parsimonious elimination of irrelevant candidates.

The overwhelming majority of the created features (156 of 162) are interval-based and return the percentage of data points in the stretch that are out of a certain range. The feature values are obtained by counting the number of data points of a certain slice that are out of the range (Center $\pm$ Multiplier * Reach) and then dividing it by the window size. The different values for the various parameters can be found in Table III below and the 156 features are the results of the combination of them (108+36+12). These interval features are named according to the schema $Center\_MultiplierReach\_mode$. The *mode* can be 'value' or 'diff' as the features are extracted from the Original Values observed or from the Differences between these, respectively.

TABLE III: PARAMETERS & EXPECTED BEHAVIOR OF INTERVAL FEATURES

Center	Reach	Multiplier	Value_in_anomalies
meanTot mean_medianTot medianTot mean_maxTot meanAfter	sd_meanTot sdBefore sdWind	3,4,5	Higher
mean_medianBef medianAfter meanAfterwind	sd_maxTot	1,2,3	Higher
mean-medianWind medianAfterwind meanLoc medianLoc	sdLoc	1	Lower

Note that, although the presented 'General Features' are scale dependent, the intervals formed are not, satisfying the criteria to be admitted in the pool of features. Now that the interval features that compose the pool were already presented, we can indicate an additional criterion that excluded some other features from being included in the pool. A set of features resulting of a certain variation should not be included in the pool if, when calculating the F1-Score with the RFC(5) and the 5-NN, the features of that set perform consistently worse than the features of a highly correlated set, ceteris paribus. For example, features using *sd-medianTot* performed worse than the ones using *sd-meanTot* across all the possible combinations of values for *Center* and *Multiplier*, and so, as the absolute correlation between pairs of features was higher than 0.9, no feature using *sd-medianTot* as *Reach* was included in the pool. In the same situation were *meanBefore* and

meanWind when compared with *meanAfter* and *meanAfterwind*, as the inclusion of the data points of the stretch for which the feature is being extracted has a positive impact in the feature performance.

The different values that the parameter *Center* can take, lead to the disposal of the interval in different levels, allowing a higher coverage of the distribution of the observations. For example, when *Center* is calculated based on a certain average, the interval has tendency to be positioned in a higher level than when a *median* based *Center* is used. This happens because the overwhelming majority of the anomalies are the result of sudden increases followed by similar decreases and their inclusion on the average pulls this measure up. When, less commonly, the anomaly is the result of a deep valley, the average is pulled down by the anomalies if we are considering the Original Values but pulled up if we are considering the absolute value of the Differences between consecutive data points. The inclusion of variations of *Center* that only consider the previous 120 data points (with the addition or not of the 24 data points of the considered stretch), i.e, the ones including *Wind*, allow to 'forget' data points that happened some time before and that could be impacting the *Center* parameter unnecessarily (e.g. when an anomaly due to a positive spike has happened a long time before but it is be pulling the *Center* up or when using the Original Values with a positive trend and the values observed a long time ago are not in the same level as the ones observed in the considered stretch, pulling the *Center* down).

On the other hand, the *Reach* parameter allow to have different amplitudes for the searching range. For *sd-meanTot*, *sdBefore* and *sdWind* the *Reach* combined with the *Center*, is defined in such a way that stretches with anomalous behavior are expected to have a higher percentage of data points falling out the interval. For example, for the simple case in which there is an easily identifiable positive spike and the remaining data points are all more or less dispersed across a lower level, these intervals are expected to have one or two data points out of the interval if the stretch considered contains information about the original data or about the differences between consecutive data points respectively. Thus, as declared in Table III, the values taken by the features derived from these *Reach* values are expected to be higher when the stretch in question is anomalous. Similarly, when *Reach* takes value *sd-maxTot* the same type of behavior is expected. However, the value that *sd-maxTot* takes is usually much higher than the remaining *Reach* possibilities as, almost always, this value corresponds to the standard deviation observed in an anomalous stretch. Thus, the *Multipliers* combined with this option take lower values. Features that take *sd-maxTot* as *Reach* are useful to avoid the identification of stretches containing small spikes as anomalous when the TS has a much bigger abnormal spike. In opposition, when the *Reach* is *sdLoc* the number of data points expected to fall out of the interval when the stretch is abnormal is lower. This happens due to the low standard deviation of a normally

behaved stretch when compared to the standard deviation of an abnormal stretch, leading to small range intervals, incapable of including a lot of data points.

TABLE IV: FORMULA & EXPECTED BEHAVIOR OF REMAINING FEATURES

Feature	Formula	Value_in_anomalies[a]
meanLoc_vs_amplLoc	$\frac{(meanLoc-(maxLoc-amplLoc))}{amplLoc}$	Further from 1/2 (-)
medianLoc_vs_amplLoc	$\frac{(medianLoc-(maxLoc-amplLoc))}{amplLoc}$	Further from 1/2 (-)
meanLoc_vs_maxLoc	$\frac{meanLoc}{maxLoc}$	Further from 1/2 (-)
medianLoc_vs_maxLoc	$\frac{medianLoc}{maxLoc}$	Further from 1/2 (-)
skewLoc	——	Further from 0 (+)
skewLoc_vs_skew-meanTot	$skew\_meanTot - skewLoc$	Further from 0 (+)

[a] Between parenthesis is represented if the value of the feature is expected to be higher or lower than the presented reference value for normal behavior

The remaining 6 features included in the pool are introduced in Table IV above. The values of the first four features presented in the table are expected to be closer to ½ when the stretch is normal as when there is no anomalous behavior in the stretch the median/mean of the data points is expected to be as far from the minimum point as from the maximum point. Once again, as most of the anomalies are the result of high positive spikes in the data, when the stretch is abnormal these features are expected to take values further from ½ with tendency to be closer to 0. When considering stretches of the differences this behavior is observed both to positive and negative spikes.

The remaining 2 features are expected to assume values further from 0 when the stretch is anomalous as in a normal behaved stretch the skewness is expected to be close to 0 and to be similar to the skewness of the majority of the remaining stretches of the TS and so similar to their average. Once again, for abnormal slices, due to the nature of the anomalies studied, the tendency of these features is to assume values higher than 0.

To confirm the discriminative power of the proposed features, the median, average, minimum, and maximum values taken by the features across normal and anomalous stretches were obtained. Naturally, the mean and the median returned were consistent to what was stated in Tables III and IV. Nevertheless, the maximum and the minimum presented no real differences between normal and abnormal behavior for most of the features. Notwithstanding, this is exactly the reason why we intend to combine several features, so that the stretches misclassified by one attribute could be correctly classified by the others. Furthermore, as stated in the Section 3, there are some annotations (corresponding to the real TS) that are subjectively classified and so can be 'wrongly' annotated, leading to this bad behavior of the features in some stretches. Moreover, we addressed the existence of a residual number of anomalies resulting from the stability in highly variable TS that can also be conducting to this behavior.

Although both alternatives will be tested, we believe that the pool of features will return better results when applied to the stretches containing information regarding the differences between data points in comparison to those which include the original data. As an alternative to the pool of engineered features, we will test the performance of an automatic feature extraction methodology that will serve as the benchmark in our study. The Time Series Feature Extraction Library (TSFEL) introduced by Barandas et al. (2020) extracts over 60 different features from the statistical, temporal, and spectral domains. The complete list of features from which the library extracts the ones considered more appropriate is available online[4].

4.3 Feature & Classifier Selection

Now that we have a set of feature vectors, one for each stretch of the training set, we shuffle it to create a homogenous dataset. Next, we need to select the features that, together, confer a better discriminative power to the model along with the classifier that is more capable of identifying the anomalies with accuracy. To perform this selection, we will establish a supervised learning problem using different classifiers and features with the goal of classifying the training data in the best way possible according to F1-Score. We will use the F1-Score instead of accuracy as in classification problems of anomaly detection, like ours, the set of observations tends to be heavily imbalance. If we decided to use accuracy, the model could be labelling all observations as normal, which is clearly incorrect, and still be designated as a good model because it is correctly identifying all the normally behaved observations that are highly represented. On the other hand, F1-Score takes into consideration both the recall and the precision, i.e., the model capacity to correctly classify both the anomalies and the normal observations.

Bota (2018) stated, based on previous literature in the study field of Human Activity Annotation, that the best classifier is the RFC and so they used this classifier to perform feature selection. Then, different classifiers were compared using the selected features and it was concluded that the RFC was, in fact, the one with a better performance. Nevertheless, we do not consider this the most correct procedure as some bias was introduced by the use of RFC to select the features in the first hand. Nothing guarantees that, if other classifier had been initially used to select the features, a different set of features, more favorable to that same classifier than to the RFC, could not be selected. Thus, to remove this bias, we will compare the performance of combinations of classifiers and features selected directly by them and select the ensemble that yields a higher F1-Score.

In the previous subsection, we stated that our pool has a lot of features that are the

[4]https://tsfel.readthedocs.io/en/latest/descriptions/feature_list.html

result of small variations from others. Therefore, it is expectable that many of these features could be highly correlated and so we must drop them. Nevertheless, the main reason behind the creation of similar features was to maximize the discriminative power through these small variations and, by dropping the correlated features without having in account their performance in the classification of the anomalies, we would not be taking advantage of that intention. Thus, we developed a small algorithm that, after calculating the correlation matrix for all features, instead of always dropping the feature that is in the column or in the row when two features are highly correlated, eliminates the one that yields a lower F1-Score when singly used to label 20% of the training set after being trained, with the pre-specified classifier, in the remaining 80%. Hence, this algorithm receives as input a set of feature vectors, a classifier, a threshold above which two features are considered highly correlated and a seed to be able to reproduce the 80/20 split of the training data. In this study, we will consider highly correlated features, a pair of features that have an absolute correlation higher than 0.9.

After eliminating all the highly correlated features, we intend to select the subset of them that, according to a certain classifier, is more capable of differentiating the anomalies from the normal behavior. To execute this task, we will use the Sequential Forward Selection (SFS) of the Mlxtend library (Raschka, 2016). This library allows the user to select a range for the number of wanted features and returns which are the best features for those pre-specified set sizes, as well as the metric obtained by those sets. To ensure the validity of the comparison between classifiers we will use the same values for the remaining parameters of the function. Thus, we decided to perform forward selection and to use a 10-fold CV to produce a more accurate estimate of the out-of-sample F1-Score. We chose this process instead of backward elimination because we believe that the number of features needed to obtain good results is not high. Hence, we set the number of features to be selected to be between 1 and 15. We decided not to use a stepwise approach because it is more costly, and we do not believe that it would boost the performance significantly. When the F1-Score for the set with x+1 features is not at least 0.1 percentage points higher than the same metric for the set with x features, we conclude that the ideal feature vector for the pre-defined classifier is the one with x features. This criterion is used to avoid the overfitting of the training data. We will test the classifiers listed in Table V below.

After, to discuss the trade-off between the model cost and performance, we will investigate if increasing even more the number of estimators of the classifier leads to an improvement in the results. In the same line of thought, we will examine if increasing the number of features in the initial pool by considering features extracted both from the Original Values and the Differences conduct to better results. For the remaining of the paper, we will interchangeably use the terms stretches and observations.

TABLE V: CLASSIFIERS

	N_estimators	Library
RFC	5,10,20	sklearn.ensemble.RandomForestClassifier
KNN	1,5,10	sklearn.neighbors.KNeighborsClassifier
SVM	—	sklearn.svm.SVC
GB	10,20	sklearn.ensemble.GradientBoostingClassifier
$XGBoost$	10,20	xgboost.XGBClassifier

4.4 Semi Supervised Active Learning Model

After establishing the ensemble of features and the classifier that yield a higher F1-Score in the Supervised classification of the training set, we may develop our model. Firstly, we have to apply the preprocessing pipeline described in Table II with the features selected by the process described in subsections 4.2 and 4.3. Finally, and based on these features and on the classifier that originated their selection, we will develop a SSAL algorithm capable of correctly classifying the stretches as normal or abnormal in a scenario in which only a small percentage of these have been labelled. To accomplish this, the model will alternate between a SSL and an AL segments. Starting from a small number of labelled observations from the training set, in each iteration of the algorithm, we intend the enlarge the number of labelled observations from this set. At the end of each iteration, the classifier is trained with the enlarged set and the model performance is tested in the holdout set. The objective is to maximize the F1-Score while reducing the number of expertly labelled observations. The process is schematized in Figure 4.

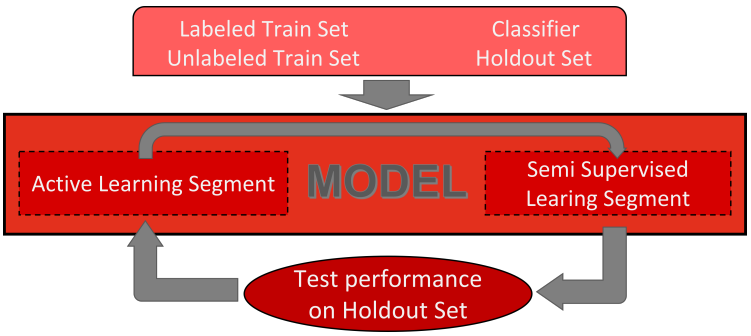

FIGURE 4: High level representation of the proposed model

Regarding the SSL segment of the model, we decided to implement a ST strategy, in which the classifier is trained with a small number of labelled observations and labels the unlabelled samples that belong to a certain class with a confidence above a certain threshold (that will be defined according to the selected classifier). This segment of the model reduces the number of times that the expert is consulted as it enlarges the labelled

training set at a much higher rate. One tested strategy in this segment is the Recursive Labelling of unlabelled samples by repeating the ST loop until no additional training observation is labelled and just then, consult an expert in the AL segment. Theoretically, this approach reduces the number of times that the expert is consulted as more observations are automatically classified per iteration. However, the iterative classification of observations without expert consultation could lead to a higher probability of misclassification of some observations. Alternatively, we will test the performance of the model when only one loop of automatic label attribution is performed per expert consultation.

The AL segment is composed by the QS, that aims to select the observation that should be labelled and incorporated into the classifier's training set based on its relevance, and by the Oracle, that is the phase where the expert is consulted to annotate the selected observation. Concerning the QS, two popular strategies are the Query by Committee Sampling (Freund et al., 1997) and the Pool-based Sampling (Kanamori, 2007). For the first, a voting system for the label of each sample is created from a committee of classifiers, being the sample with higher label disagreement the one selected as most informative to be labelled. In the second strategy, a pool of the unlabelled samples is created and the QS selects the most informative sample to be labelled according to a pre-defined metric (F1-Score for the purposes of this study). As Bota (2018) stated that Pool-based Sampling tends to obtain better results, we decided to follow this strategy.

As baseline method, we will use Random-based QS (R-QS), against which we will test the performance of other QS. The first alternative is the widely spread Uncertainty-based QS (Unc-QS). Bota (2018) discussed the use of different strategies to calculate the uncertainty of the samples, namely Least Confident Sampling, Margin Sampling and Entropy Sampling. Notwithstanding, as we are facing a binary classification problem, all these strategies return the same output. The idea is to select the observation for which the classifier is less certain about, i.e., the observations for which, according to the pre-trained classifier, the probabilities of belonging to the anomaly or to the normal class are closer (and so closer to 0.5). However, the use of simple classifiers leads to the obtainment of equally uncertain observations. Among these is selected only one to be annotated, introducing some randomness to the process. Besides, uncertainty based QS tend to select samples closer to the decision boundary, which although helping classifiers such as the SVM, whose objective is to maximize the hyperplane margin between decision boundaries, tend to neglect the prior feature distribution space and to introduce sampling bias when other classifiers are used. Additionally, this QS may select isolated observations (due to low similarity with the classes), which do not constitute representative data.

As an alternative, we will try a Utility-based QS (Ut-QS), selecting the observation

that is in a region with higher density. To achieve it, we will calculate the distance between all pairs of observations. To avoid the invariable selection of normally behaved observations as these are in higher density regions, we decided not to use the distance between observations directly as the measure of utility. Thus, the utility measure is defined as the number of unlabelled observations that are closer to the considered observation than to any other of the unlabelled set.

Combining these two strategies, we developed a QS in which between all the observations that have the higher uncertainty, selects the one that has a higher utility, highly reducing the number of observations for which the utility must be calculated (Unc&Ut1-QS). We also created an alternative in which instead of, in the first phase, only selecting the observations with the higher value of uncertainty, selects the samples that have higher values of uncertainty until at least 20% of the unlabelled set is selected and then performs a linear combination with the same weights for the values of uncertainty and utility obtained, selecting the annotation that returns a higher final value (Unc&Ut2-QS).

The Oracle decision regarding the classification of the observation selected to be tagged will be substituted by the querying of the hidden label of it, simulating a situation in which we only have access to the real label when consulting the expert.

The SC is activated when at least 50% of the training data is labelled and the expert is 5 or more times consulted without resulting in any automatically classified observations in the training set during these loops. The 50% threshold is used to avoid the process stoppage when the number of tagged observations is low, which can happen when initializing the model with a reduced number of observations that do not confer sufficient information for the automatized labelling part. The process is schematized in Figure 5, with the enlargement of the labelled training set before every time that the classifier is trained and with the unfilled arrow representing an optional path.

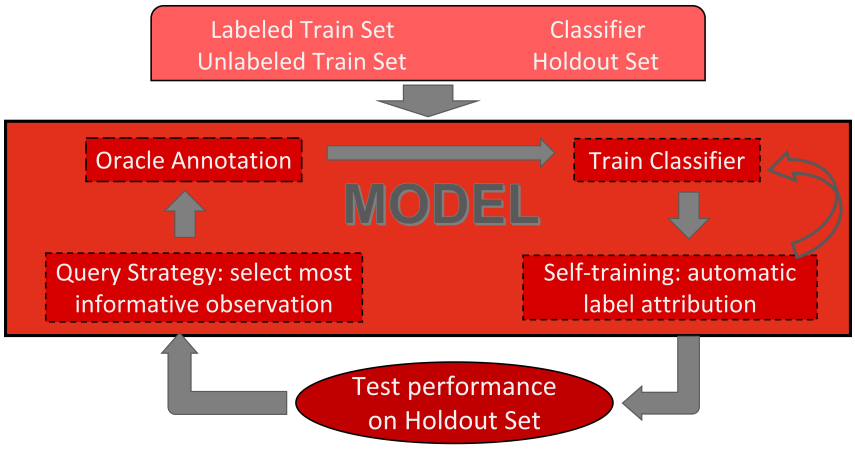

FIGURE 5: More detailed representation of the proposed model

5 RESULTS

As described in the previous section, we need to select the ideal feature vector to develop the SSAL model. After applying the proposed process to separate the data into training and holdout sets, 27 and 6 initial data points were removed from the 330 and 37 TS that constitute the sets, respectively, as they were considered irregularities in the beginning of the TS. Succeeding the division of the TS into stretches, we need to confirm the compatibility of the separation of the holdout set with the methodology used to obtain the stretches. Table VI shows the quantity of anomalies in the different sets and the total number of stretches obtained in the presence and in the absence of overlap use.

TABLE VI: ANOMALY DISTRIBUTION ON DIFFERENT SETS

	Overlap			No Overlap		
	Train	Holdout	Total	Train	Holdout	Total
# Anomalous Stretches	3746	437	4183	1892	223	2115
# Stretches	42660	4795	47455	21551	2422	23973
% of Anomalies	8.78%	9.11%	8.81%	8.78%	9.21%	8.82%

As for both cases, the percentage of anomalies is more or less similar across sets, the assumption about the validity of the combination of these processes is confirmed. Hence, the data used for training the classifier is identical to the one used to test the results, which is essential to ensure that an eventual bad performance of the model does not result from the difference between sets. We can also corroborate the imbalance of the observations, justifying the use of the F1-Score as the performance metric. The performance of different classifiers with regard to this measure is exposed in Table VII.

TABLE VII: CLASSIFIERS PERFORMANCE

	Original Values	Differences	Differences+No_overlap
$TSFEL + RFC20$	**90.70% (13)**	90.80% (7)	90.82% (7)
$TSFEL + XGBoost20$	**89.88% (11)**	90.90% (8)	91.05% (8)
$1 - NN$	88.44% (20)	92.44% (12)	92.56% (12)
$5 - NN$	84.08% (12)	93.80% (10)	92.85% (7)
$10 - NN$	78.29% (10)	93.47% (11)	92.62% (6)
SVM	66.67% (8)	93.12% (9)	92.64% (7)
$RFC5$	**90.17% (17)**	**94.40% (13)**	93.31% (10)
$RFC10$	**90.13% (15)**	**94.53% (11)**	93.33% (9)
$RFC20$	**90.76% (17)**	**94.82% (11)**	93.54% (9)
$GB10$	61.29% (5)	91.70% (6)	91.28% (5)
$GB20$	66.92% (4)	92.39% (6)	92.72% (7)
$XGBoost10$	84.45% (15)	**94.25% (9)**	93.63% (6)
$XGBoost20$	87.58% (21)	**94.03% (7)**	93.79% (7)

The format of the results is as follows: F1-Score (number of features) obtained by the classifier when the SC was activated.

There, it is compared the selection of the ideal feature vector from the discussed pool of 162 features, with 11 alternatives for classifier, against the selection from the automatically extracted features by the TSFEL. Furthermore, we compare the extraction of this features from the Original Values and from the Differences obtained and we study the use of overlap in the division of TS.

Before performing the forward selection of the ideal feature vector, the correlated features with lower F1-Score according to the pre-specified classifier were dropped. As stated while engineering the features, many of them are slight variations from others with the goal of maximizing the discriminative power, and so, depending on the classifier, the number of dropped features is between 105 and 111, leaving between 51 and 57 candidates for the SFS. The evolution of the F1-Score as more features are added to the ideal feature vector, until the SC is met, can be found in Figure A.3.

In the Table above we can see that, using the Original Values, the automatically extracted features (with both tested classifiers) perform similarly to the best classifiers that use the proposed features. In fact, to obtain similar results, the ideal feature vector needs a higher number of proposed features than when these are automatically extracted. Surprisingly, the SVM and both variations of the Gradient Boost perform poorly in this scenario.

Not so surprisingly, the performance when using the Differences instead of the Original Values is better across all classifiers. For the proposed features, this is easily justifiable by the uniform behavior regarding positive and negative spikes and by the duplication of 'extreme' data points, as for each spike in the Original Values, we have two high values corresponding to the increase before the data point and to the decrease posterior to this one, or vice-versa, conceding a higher discriminating power to the features. In this scenario, the proposed features perform better than the automatically extracted for all classifiers, proving the assumption that expertly selected features tend to perform better. The best classifier, as when using the Original Values, is the RFC with 20 decision trees.

In the absence of overlap, although the slightly better performance for the TSFEL features and for a pair of classifiers for the proposed features, the overall performance seems to be worse than when using 50% overlap to obtain the slices. This might be due to the division of the TS in inconvenient points of time, as discussed before, or due to the lower number of stretches obtained while using this technique, as there is a reduction in the number of stretches used for training the classifier. Objectively, without using overlap, there was no classifier capable of surpassing the barrier of the 94%. On the other hand, this threshold is surpassed by the XGBoost (with a lower number of features) and RFC classifiers while using the Differences with overlap. Hence, in Figure 6, the performance of these classifiers with the increase of the length of the feature vector is plotted.

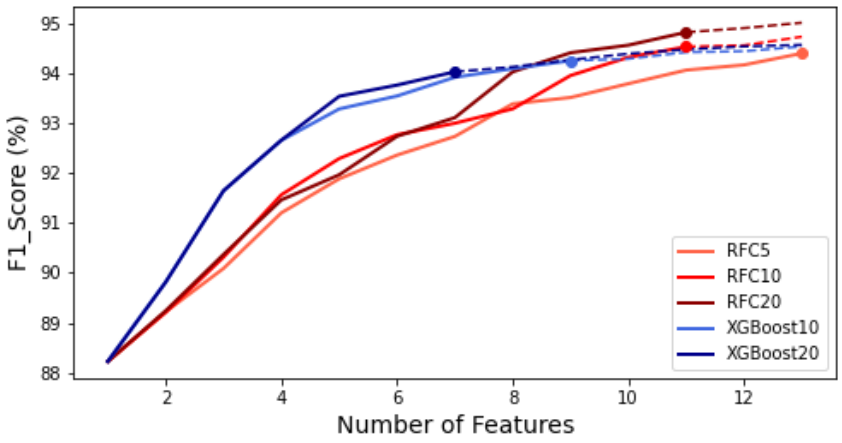

FIGURE 6: Classifiers: F1-Score vs Number of Features

Although initially the F1-Scores of the XGBoosts increase faster, these performances are surpassed by the RFC with 20 and 10 estimators for more than 9 and 11 features respectively. Note that, for the 5 classifiers analysed, the F1-Score with just one feature is equal. This suggests that the first feature selected is the same for all classifiers.

TABLE VIII: FIRSTLY SELECTED FEATURES BY CLASSIFIER

	RFC5	RFC10	RFC20	XGBoost10	XGBoost20
$meanAfterwind\_5sd - meanTot\_diff$	**1st**	**1st**	**1st**	**1st**	**1st**
$meanLoc\_1sd - maxTot\_diff$	2nd		13th	8th	7th
$medianLoc\_5sdWind\_diff$	3rd				11th
$mean - medianTot\_1sdLoc\_diff$	4th	11th		4th	4th
$meanLoc\_1sdLoc\_diff$	5th	10th	14th		12th
$mean - medianBef\_1sd - maxTot\_diff$		2nd	2nd	6th	
$meanAfterwind\_1sdLoc\_diff$	14th	3rd	4th		
$mean - maxTot\_4sd - meanTot\_diff$	6th	4th	5th	3rd	3rd
$meanTot\_1sdLoc\_diff$		5th	6th		
$meanLoc\_5sdWind\_diff$			3rd		
$skewLoc\_vs\_skew - meanTot\_diff$	7th	8th	7th	2nd	2nd
$medianLoc\_3sdWind\_diff$		15th		5th	5th

Only listed the features that are between the first 5 selected for at least on of the classifiers. Only considered positions 1st to 15th. Underlined positions mean that the feature is not included in the ideal feature vector.

Table VIII above demonstrates that *meanAfterwind_5sd-meanTot_diff* is unanimously the feature with a higher discriminating power. In fact, these 5 classifiers trained solely on this feature perform fairly well, producing a F1-Score of 88.22%. Moreover, when using the Differences and with the exception of the 1-NN with no overlap, all the classifiers firstly select this feature reaching performances around 88%. To achieve similar results, there are required at least 4 TSFEL features (Figure A.3). The low quantity of blank spaces in the Table evidences the coherence in the selection of features across classifiers. Note that *skewLoc_vs_skew-meanTot_diff* and *mean-maxTot_4sd-meanTot_diff* also belong to the ideal feature vectors for the 5 best classifiers. For the remaining of the study, we will use the RFC with 20 decision trees applied to the stretches of Differ-

ences between consecutive points, obtained with 50% overlap as this proved to be the best scenario. Thus, Table IX lists the 11 features selected by this classifier.

TABLE IX: IDEAL FEATURE VECTOR (IN ORDER)

	Feature
$1st$	meanAfterwind_5sd-meanTot_diff
$2nd$	mean-medianBef_1sd-maxTot_diff
$3rd$	meanLoc_5sdWind_diff
$4th$	meanAfterwind_1sdLoc_diff
$5th$	mean-maxTot_4sd-meanTot_diff
$6th$	meanTot_1sdLoc_diff
$7th$	skewLoc_vs_skew-meanTot_diff
$8th$	mean-maxTot_1sdLoc_diff
$9th$	medianLoc_3sdBefore_diff
$10th$	mean-maxTot_3sd-meanTot_diff
$11th$	mean-medianWind_3sd-meanTot_diff

Firstly, we applied a supervised model, trained using all the training data available and tested in the holdout set. At this point, a new question was raised as the results obtained while utilizing the features in the order that they were selected is different then when using the order returned by the SFS (alphabethical order). Although this was expected as, even using the same seed, the RFC does not ensemble the same decision trees, the obtained results should be similar. This is what happens for the proposed feature vector that produces the similar F1-Scores of 93.33 and 92.69 when the selection and the alphabetical order are utilised, respectively, and proving to generalize satisfactorily. The same can not be said for the TSFEL features that return 90.67/11.35 for the RFC and 90.97/11.64 for the XGBoost, showing to be highly dependend of the order and so generalizing poorly. For the remaining of the paper, we decided to utilize the order returned by the SFS as this proved to enforce the assumption of quality in the feature selection. Now, we will analyse if the proposed SSAL model permits a significative reduction of the labelled training observations, without having a substantially negative impact in the performance.

Due to the good performance obtained with a low number of features and as the RFC is a simple classifier, we believe that the classifier will be capable of easily enlarging the training set. Thus, to avoid the misclassification of observations, these should only be labelled if the classifier is completely certain about it, i.e., if all the decision trees agree with the classification. Therefore, in a first instance, we will initialize the model merely with 40 randomly selected observations, divided into 36 normally and 4 abnormally classified to reproduce the proportions expected. Note that 40 observations represent less than 0.1% of the training data. In Figures 7 and 8, we compare the suggested QS using Recursive Labelling (RL) or Non-Recursive Labelling (NRL) in the SSL segment, respectively.

Table X summarizes the results obtained for both situations.

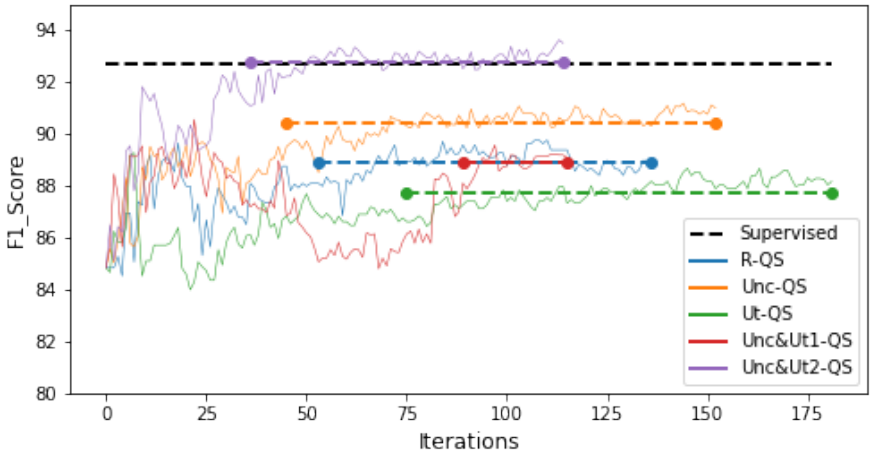

FIGURE 7: Recursive Labelling with 40 initial observations

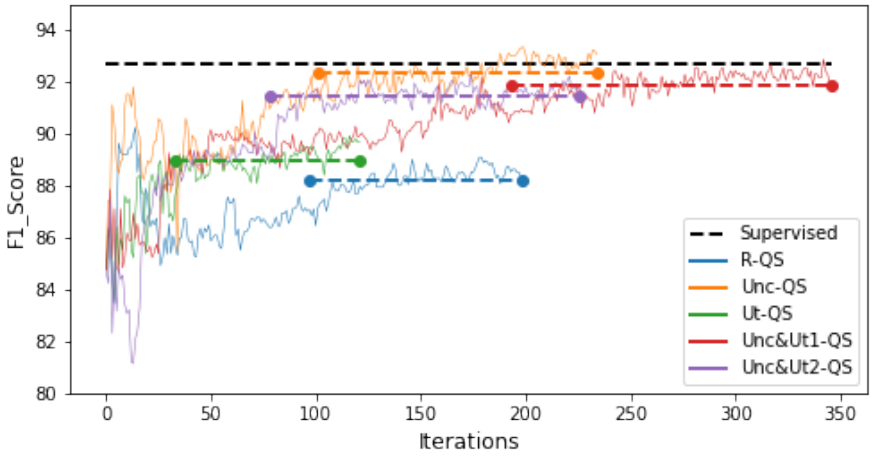

FIGURE 8: Non-Recursive Labelling with 40 initial observations

TABLE X: QUERY STRATEGY PERFORMANCE WITH 40 INITIAL OBSERVATIONS

		R-QS	Unc-QS	Ut-QS	Unc&Ut1-QS	Unc&Ut2-QS
F1-Score	RL	88.94	91.00	88.17	89.08	**93.49**
(Final)	NRL	88.28	**93.06**	89.67	91.94	91.58
F1-Score	RL	89.76	91.17	89.27	90.55	93.62
(Max)	NRL	90.24	93.33	89.92	92.87	92.33
For	RL	327	763	266	1268	622
Classifying	NRL	219	268	369	131	199
Stability Interval	RL	[53,136]	[45,152]	[75,181]	[89,115]	[36,114]
(Iterations)	NRL	[97,198]	[101,234]	[33,121]	[193,346]	[78,226]
F1-Score	RL	88.92	90.41	87.75	88.88	92.79
(Interval Mean)	NRL	87.75	91.78	89.26	91.23	90.92

The underlined values mark the maximal F1-Scores that occur out of the stability interval and the bolt values mark the final F1-Scores higher than the one obtained with the Supervised Model: 92.69%

The figures show that the performance of the SSAL model does not monotonously increase with the iterations and, consequently, with the increasing number of training observations labelled. This is natural, as the enlargement of the training set is done by a ST algorithm that may misclassify some observations along the process. However, although not yielding the maximal F1-Score, the last iteration and, more specifically, the F1-Score returned at that point must be the comparison criterion between the alternatives, as in the absence of labels we would not recognize at which iteration the best performance occurred. In Figures 7 and 8, due to this noise on the performance metric utilized, we decided to compute an interval with low variability and the average F1-Score obtained within it. Hence this interval is composed by the range of iterations to which the standard deviation is below 0.5% with the end point being the final iteration before the SC activation. These intervals, that can be consulted in Table X, are represented in the figures by dashed lines and demonstrate that, normally, the SC is activated when the performance is not highly variable from iteration to iteration in contrast with the initial behavior.

Both for Recursive and Non-Recursive Labelling scenarios, the barrier of 91% is surpassed both by Unc-QS and Unc&Ut2-QS, with the final F1-Score of the latter for the Recursive Labelling being 0.8 percentage points better than the one yielded when using Supervised Learning. Although Unc&Ut1-QS clearly outperforms R-QS in the Non-Recursive Labelling case, the two QS achieve similar results with Recursive Labelling, being Unc&Ut1-QS less stable and so not being the greatest alternative.

Note that, for all cases, the final F1-Scores are higher than the average of this metric in the stability interval. Moreover, with the exception of the Ut-QS, when the SC is activated, there are more observations yet to be classified in the Recursive scenario. This may result of the lower control on the labelling of observations, as more observations are classified without consulting the Oracle. Hence, and as expected, the SC in this scenario is activated after a smaller number of iterations. As the Non-Recursive alternative runs for more iterations and classify a higher number of observations, it was expected to yield better results due to the higher control to avoid a domino misclassification. However, that can not be concluded, as there are QS that perform better in each of the scenarios.

Nevertheless, for both scenarios, any QS has a satisfactory performance, taking into consideration that the model was initialized with less than 0.1% of the training data labelled and that, in the worst case, there is only a decrease of 5% in comparison with the Supervised Model. However, the figures show that without consulting the expert any time, i.e., in iteration 0, the performance of the model is fairly good (84.84% and 84.77% for Recursive and Non-Recursive Labelling respectively). This happens due to the high volume of observations that the classifier is 'certain' enough to attribute a label, being 38785

and 35586 observations respectively classified before consulting the expert any time. Due to this good performance with only 40 initial observations, we decided to decrease the initialization set to only 2 randomly selected observations: one for each type of behavior.

As illustrated in Figure 9 (left), only 4 of the 5 QS addressed before are compared. The one left out is the Ut-QS. As we initialize the algorithm with only two observations labelled, the SSL segment is unable to label any observations before consulting an expert, which obligates, according to this QS, to calculate the distances between all the 42658 unlabelled observations, leading to non-competitive time and computational costs.

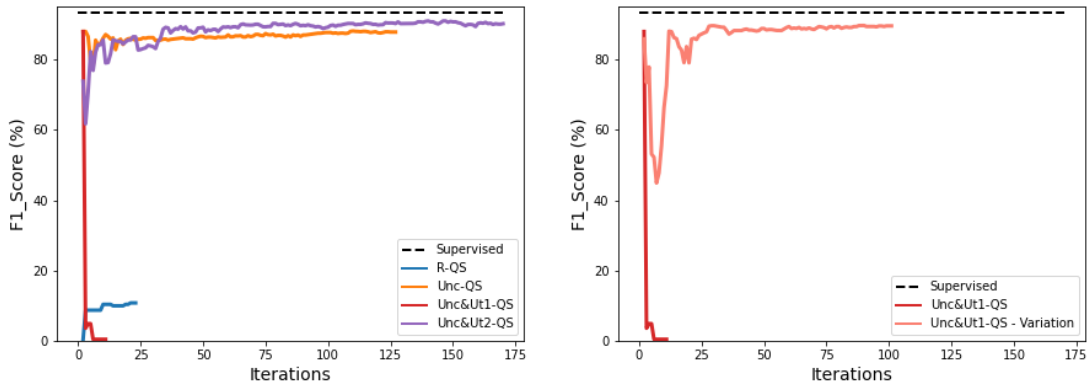

FIGURE 9: Recursive Labelling with 2 initial observations

In a situation in which the SSL segment is not capable of differentiating normal from abnormal behavior before the enlargement of the training set by the expert, the importance of the QS selection is higher. In fact, for the Recursive alternative, there is a much higher discrepancy between QS, with the R-QS performing poorly. By randomly choosing the observations to be labelled by the Oracle, we can be selecting misleading observations, such as the one presented in Figures 2 and A.2 , leading to a domino growth in the number of misclassified observations and consequently to bad performances.

Another situation that can lead to this behavior is the constant selection of observations from the same class, leading to an enlargement of the set through the addition of observations with the same label and to a tremendous imbalance of the training set that induce the misclassification of observations by wrongly attributing the label with a representation proportion above the expected. This is the case of Unc&Ut1-QS that, despite being able to perform finely after a small number of iterations, starts to fail due to the over-attribution of the normal label, obtaining a disappointing performance, as shown in Table XI.

TABLE XI: QS PERFORMANCE - RL WITH 2 INITIAL OBSERVATIONS

	R-QS	Unc-QS	Unc&Ut1-QS	Unc&Ut1-Variation	Unc&Ut2-QS
F1-Score (%)	10.80	87.80	0.46	80.43	90.18
Stopping Iteration	23	127	11	26	170
Time Spent (s)	139	181	122	164	474

This problem was previously mentioned in He et al. (2015) and so, to avoid the steep decrease of the performance, we tested a variation of Unc&Ut1-QS. In Table VI was stated that the expected percentage of anomalies is around 10%. Thus, while choosing the observations to be expertly labelled, we took into consideration the percentage of labels per class attributed until that moment. Hence, if the proportion of anomalies in the labelled set is between 5% and 15%, the Unc&Ut1-QS is normally called on the complete unlabelled set. However, if the percentage is lower than 5%, the QS tries to select an observation with higher probability of being an anomaly. To achieve it, the Unc&Ut1-QS is only called after the application of a filter to select the observations that, in the previous iteration, had a predicted probability of belonging to the anomaly class higher than 50% according to the classifier. If there are no observations in this situation, the filter selects the observations with higher probability (although lower than 50%) of being an anomaly and calls the Ut-QS on these ones, as they all have the same uncertainty. A similar process is applied when the percentage is above 15% to enable the selection of a normal observation. This variation improves the F1-Score by 80% but it is still not competitive with Unc-QS and Unc&Ut2-QS that, once again perform well. Next, we will check the performance of Non-Recursive Labelling with only 2 initial observations labelled.

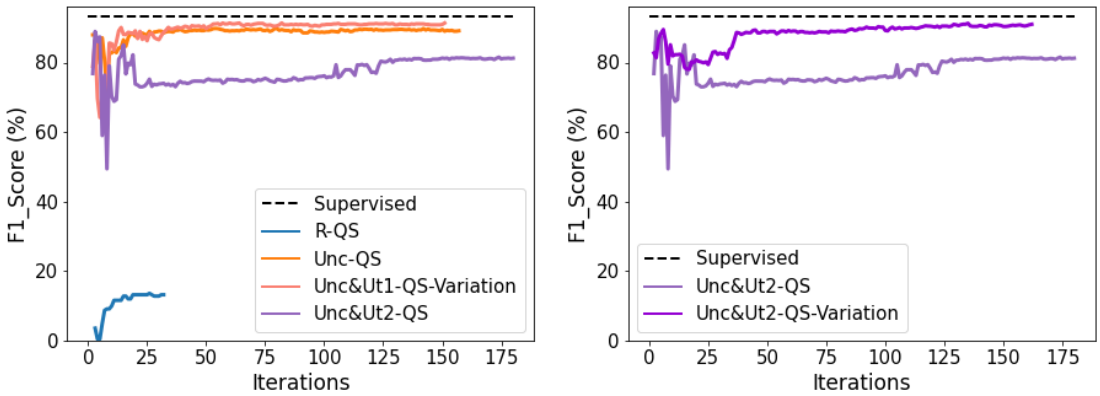

FIGURE 10: Non-Recursive Labelling with 2 initial observations

Figure 10 (left) shows that, for the Non-Recursive alternative, the R-QS also performs poorly. On the other hand, the Unc-QS and the variation of Unc&Ut1-QS perform well and it is the Unc&Ut2-QS that, for the first time, has a worse performance than expected. To solve it, we applied the variation described before to Unc&Ut2-QS and the results

are shown in Figure 10 (right). The proposed filtering of the observations increased the performance of the model, making it competitive with the other alternatives and proving to be efficient. The results obtained can be consulted in Table XII below.

TABLE XII: QS PERFORMANCE - NRL WITH 2 INITIAL OBSERVATIONS

	R-QS	Unc-QS	Unc&Ut1-Variation	Unc&Ut2-QS	Unc&Ut2-Variation
F1-Score	13.22	89.10	91.34	81.22	90.98
Stopping Iteration	32	157	151	180	162
Time Spent	136	165	187	829	363

Note that the F1-Scores of the variations of Unc&Ut1-QS and Unc&Ut2-QS surpass the barrier of the 90%, which implies a decrease below 2% when compared to the Supervised approach while having a quantity of manually labelled observations astonishingly lower. Moreover, these performances are not obtained by increasing the number of times that the expert is consulted, as for the cases that still perform finely, the stopping iteration is lower than when the initial set was composed by 40 observations. Furthermore, comparing with the results obtained for the Recursive alternative when the initialization set is just 2 observations, we concluded that using the Non-Recursive ST the model performs better as all QS yield higher F1-Scores.

Additionally, the temporal and computational costs are low for the obtained results. The preprocessing described in Table II only takes 78 seconds for the 367 original TS and the time spent in the classification is also low considering the huge decrease in the number of labelled observations and the negligible lost in the performance as shown in Tables XI and XII. Note that the filtering of the observations before the application of the Unc&Ut2-QS permits the halving of the time spent because in the original QS it was necessary to calculate the utility for a higher number of observations.

Due to the celerity of the discussed alternatives, we decided to test if an expansion of the initial pool of features conjugated with an increase in the number of estimators could increase even more the performance of the proposed method. To investigate it, we augmented the pool of features by joining the features extracted from the Differences and from the Original Values. By using the 2 classifiers that performed better before and changing their number of estimators, we obtained the results shown below on Table XIII.

The best performance is obtained by the XGBoost with 25 estimators, followed by the RFC with the same number of decision trees. Before applying the SSAL model, we needed to check if the selected features generalize well, by performing a Supervised Classification. A performance of 94.52% was obtained, being 1.93% higher than when solely using the Differences. It remained to understand, if the selected features performed better than when using the Differences in a situation in which the amount of labels is small and if the cost of that hypothetical improvement justified its application, in a trade-off between

performance and cost. The temporal cost may increase due to the increment in the number of estimators and/or by the prolongation in the obtainment of the 'General Feats' that are extracted from two different TS, namely the Original Values and the Differences.

TABLE XIII: CLASSIFIERS' PERFORMANCE - NEWPOOL

	F1-Score
$RFC20 - Differences$	94.82% (11)
$XGBoost10 - Differences$	94.25% (9)
$RFC20 - Newpool$	95.15% (11)
$RFC25 - Newpool$	**95.42% (12)**
$RFC30 - Newpool$	95.40% (12)
$XGBoost10 - Newpool$	95.27% (12)
$XGBoost20 - Newpool$	95.39% (11)
$XGBoost25 - Newpool$	**95.51% (11)**
$XGBoost30 - Newpool$	95.38% (10)

The format of the results is as follows: F1-Score (number of features) obtained by the classifier when the SC was activated.

Hence, we applied our SSAL model with the feature vector selected by the XGBoost25 and as 'complete certainty' is rarer than when using the RFC, we decreased the probability threshold above which an observation may be tagged from 1 to 0.97. Nevertheless, the XGBoost classifier was unable to perform any prediction when the initialization set was composed by only two observations, attributing probability 0.5 of belonging to either one of the classes for all observations. Consequently, all observations presented the same uncertainty, leading to a completely random selection when the Unc-QS is applied. To make things worse, the 2 proposed methods that take into consideration both the uncertainty and the utility have unfeasible temporal costs, as the utility of all observations, and so the distance between all pairs of observations (except the two labelled ones) must be calculated, making these alternatives non-competitive.

As an alternative, we could implement a random selection in the first iterations, followed by the normal application of the proposed QS when the classifier is capable of making predictions. However, we demonstrated before the disastrous results of randomly selecting the observation to be tagged when the set of labelled observations is small.

Unlike the XGBoost, we already demonstrated that the RFC is capable of predicting with only two observations and so we applied a Supervised Classification with its selected features. We obtained a performance of 87.85% which represents a loss of 4.84 percentage points to the model using the Differences. As we are increasing the number of decision trees and so, decreasing the probability of overfitting, we concluded that this incapacity to generalize must be due to the selected feature vector and so it makes no sense to apply our SSAL model based on it.

6 CONCLUSION

This study aimed to compare the use of expertly proposed and automatically extracted features to perform outlier-type anomalies detection. Feature vectors obtained from the proposed initial pool of features accomplished better results than the ones obtained with the use of the TSFEL, mainly in the generalization part where there is a discrepancy of around 81% in the F1-Score returned in favor of the proposed features.

Using the ideal feature vector, it was developed a SSAL model capable of performing in a situation of label scarcity, with different QS and ST methodologies compared. When initializing the model with only 40 observations, which represent less than 0.1% of the training data, all QS perform satisfactorily as the first iteration of the ST tags a huge number of observations before consulting the expert. In fact, the QS that takes into consideration both the uncertainty and the utility of the unlabelled observations outperforms the Supervised alternative.

When the initial set of labelled observations is reduced to only 2 observations, the importance of the QS increased, with randomly selecting the observations to be tagged proving not to be a viable alternative. Variations of the QS that take into consideration the percentage of anomalies in the training set outperformed the QS that did not filter the observations to be labelled. It was concluded that QS based on the uncertainty and the utility perform only 2 percentage points worse than the Supervised alternative with an astonishingly lower number of labels.

We noted that some tagged data points in the used dataset were not consistent, being almost all anomalies the result of high variation in the TS and a much smaller set of them the result of low variation in highly variable TS, which can be affecting the performance of the model. In the future, we believe that a new label should be introduced to distinguish these types of abnormal behavior. A future objective will be to test a SC based on the low classification change of the holdout set, i.e., it would be activated when the percentage of changes in the labels attributed in this set is small.

REFERENCES

Aggarwal, J. K. (2005), Human activity recognition, *in* 'International Conference on Pattern Recognition and Machine Intelligence', Springer, pp. 39–39.

Au, C. E., Skaff, S. and Clark, J. J. (2006), Anomaly detection for video surveillance applications, *in* '18th International Conference on Pattern Recognition (ICPR'06)', Vol. 4, IEEE, pp. 888–891.

Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T. and Gamboa, H. (2020), 'Tsfel: Time series feature extraction library', *SoftwareX* **11**, 100456.

Bastos, J. A. and Caiado, J. (2021), 'On the classification of financial data with domain agnostic features', *International Journal of Approximate Reasoning* **138**, 1–11.

Berndt, D. J. and Clifford, J. (1994), Using dynamic time warping to find patterns in time series., *in* 'KDD workshop', Vol. 10, Seattle, WA, USA:, pp. 359–370.

Blum, A. and Mitchell, T. (1998), Combining labeled and unlabeled data with co-training, *in* 'Proceedings of the eleventh annual conference on Computational learning theory', pp. 92–100.

Bota, P. J. (2018), Human Activity Annotation based on Active Learning, PhD thesis.

Braei, M. and Wagner, S. (2020), 'Anomaly detection in univariate time-series: A survey on the state-of-the-art', *arXiv preprint arXiv:2004.00433* .

Chang, I., Tiao, G. C. and Chen, C. (1988), 'Estimation of time series parameters in the presence of outliers', *Technometrics* **30**(2), 193–204.

Chu, S., Keogh, E., Hart, D. and Pazzani, M. (2002), Iterative deepening dynamic time warping for time series, *in* 'Proceedings of the 2002 SIAM International Conference on Data Mining', SIAM, pp. 195–212.

Chuah, M. C. and Fu, F. (2007), Ecg anomaly detection via time series analysis, *in* 'International Symposium on Parallel and Distributed Processing and Applications', Springer, pp. 123–135.

Cohen, I., Cozman, F. G., Sebe, N., Cirelo, M. C. and Huang, T. S. (2004), 'Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(12), 1553–1566.

Freund, Y., Seung, H. S., Shamir, E. and Tishby, N. (1997), 'Selective sampling using the query by committee algorithm', *Machine learning* **28**(2), 133–168.

Fulcher, B. D. (2017), 'Feature-based time-series analysis', *arXiv preprint arXiv:1709.08055* .

Fulcher, B. D. and Jones, N. S. (2014), 'Highly comparative feature-based time-series classification', *IEEE Transactions on Knowledge and Data Engineering* **26**(12), 3026–3037.

Ganesalingam, S. and McLachlan, G. (1978), 'The efficiency of a linear discriminant function based on unclassified initial samples', *Biometrika* **65**(3), 658–665.

He, G., Duan, Y., Li, Y., Qian, T., He, J. and Jia, X. (2015), Active learning for multivariate time series classification with positive unlabeled data, *in* '2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)', IEEE, pp. 178–185.

He, G., Li, Y. and Zhao, W. (2017), 'An uncertainty and density based active semi-supervised learning scheme for positive unlabeled multivariate time series classification', *Knowledge-Based Systems* **124**, 80–92.

Itakura, F. (1975), 'Minimum prediction residual principle applied to speech recognition', *IEEE Transactions on acoustics, speech, and signal processing* **23**(1), 67–72.

Kanamori, T. (2007), 'Pool-based active learning with optimal sampling distribution and its information geometrical interpretation', *Neurocomputing* **71**(1-3), 353–362.

Keogh, E. J. and Pazzani, M. J. (2000), Scaling up dynamic time warping for datamining applications, *in* 'Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 285–289.

Keogh, E. and Kasetty, S. (2003), 'On the need for time series data mining benchmarks: a survey and empirical demonstration', *Data Mining and knowledge discovery* **7**(4), 349–371.

Keogh, E., Lin, J. and Fu, A. (2005), Hot sax: Efficiently finding the most unusual time series subsequence, *in* 'Fifth IEEE International Conference on Data Mining (ICDM'05)', Ieee, pp. 8–pp.

Keogh, Y. C. B. H. E. and Batista, G. E. (2013), 'Dtw-d: Time series semi-supervised learning from a single example'.

Liang, Z., Wang, H., Ding, X. and Mu, T. (2021), 'Industrial time series determinative anomaly detection based on constraint hypergraph', *Knowledge-Based Systems* p. 107548.

Lorenz, J., Silva, M. I., Aparício, D., Ascensão, J. T. and Bizarro, P. (2020), 'Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity', *arXiv preprint arXiv:2005.14635* .

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D. and Jones, N. S. (2019), 'catch22: Canonical time-series characteristics', *Data Mining and Knowledge Discovery* **33**(6), 1821–1852.

Machado, I. P. (2013), Human activity data discovery based on accelerometry, PhD thesis, Faculdade de Ciências e Tecnologia.

Müller, M. (2007), 'Dynamic time warping', *Information retrieval for music and motion* pp. 69–84.

Nguyen, M. N., Li, X.-L. and Ng, S.-K. (2011), Positive unlabeled learning for time series classification, *in* 'Twenty-Second International Joint Conference on Artificial Intelligence'.

Raschka, S. (2016), 'Mlxtend'.

Sakoe, H. and Chiba, S. (1978), 'Dynamic programming algorithm optimization for spoken word recognition', *IEEE transactions on acoustics, speech, and signal processing* **26**(1), 43–49.

Salvador, S. and Chan, P. (2007), 'Toward accurate dynamic time warping in linear time and space', *Intelligent Data Analysis* **11**(5), 561–580.

Settles, B. (2009), 'Active learning literature survey'.

Teng, M. (2010), Anomaly detection on time series, *in* '2010 IEEE International Conference on Progress in Informatics and Computing', Vol. 1, IEEE, pp. 603–608.

Tukey, J. W. et al. (1977), *Exploratory data analysis*, Vol. 2, Reading, Mass.

Wang, H., Zhang, Q., Wu, J., Pan, S. and Chen, Y. (2019), 'Time series feature learning with labeled and unlabeled data', *Pattern Recognition* **89**, 55–66.

Wei, L. and Keogh, E. (2006), Semi-supervised time series classification, *in* 'Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 748–753.

Zhu, J., Wang, H., Tsou, B. K. and Ma, M. (2009), 'Active learning with sampling by uncertainty and density for data annotations', *IEEE Transactions on audio, speech, and language processing* **18**(6), 1323–1331.

Zhu, J., Wang, H., Yao, T. and Tsou, B. K. (2008), Active learning with sampling by uncertainty and density for word sense disambiguation and text classification, *in* 'Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)', pp. 1137–1144.

A Appendices

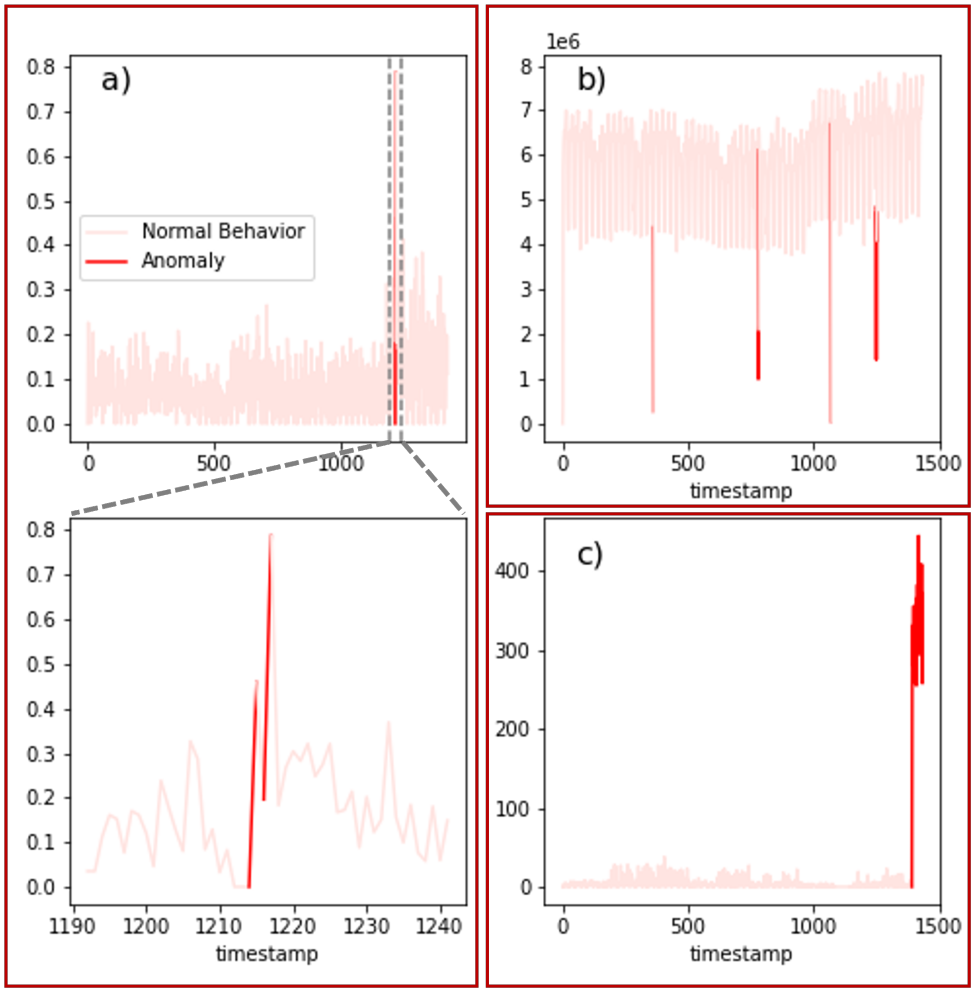

FIGURE A.1: Examples of Anomalies

a) Most common type of anomaly: sudden increase of the observed value.
b) Anomalies resulting from sudden decreases on the observed value. Note that the first valley is the initial observation and although being lower than the remaining ones, it is not tagged as an anomaly.
c) Set of data points tagged as anomalous due to a sudden increase that causes a highly variable zone compared to the previously observed values of the TS.
Note that the 3 TS present completely different scales.

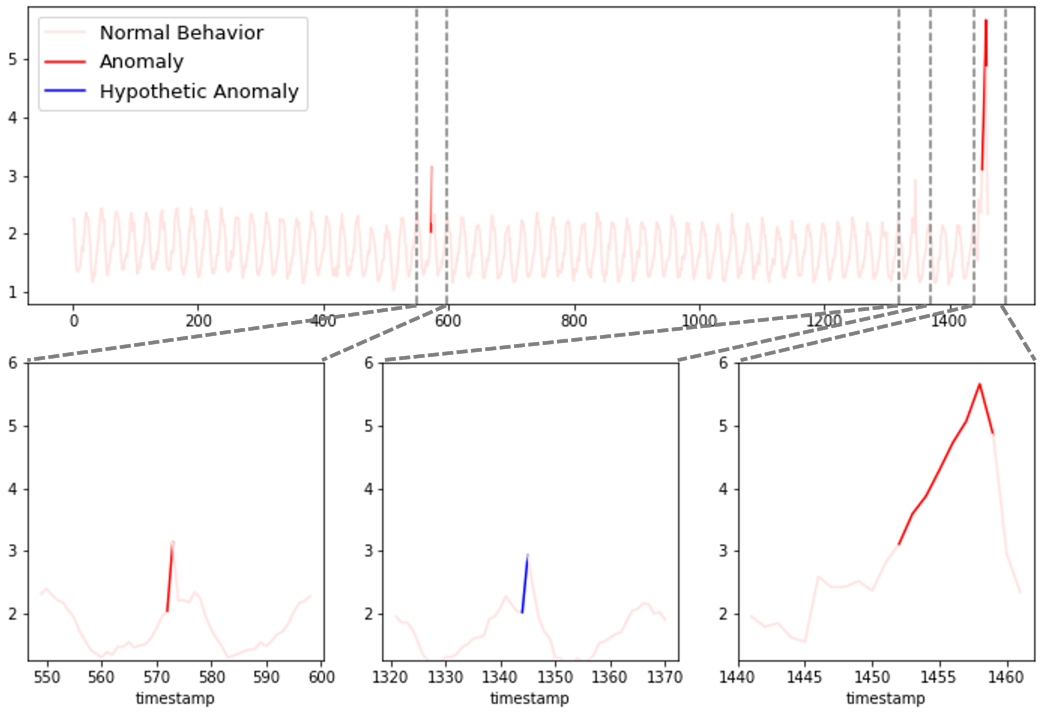

FIGURE A.2: Label attribution subjectivity

The TS presents 3 positive spikes, being the last the one with largest variation and easily identified as an anomaly. The remaining 2 peaks are quite similar. Nevertheless, the first is tagged as an anomaly, while the second is considered normal behavior, showing the subjectivity in the label attribution.

TABLE A.I: GENERAL FEATS

Features	Description
label	Takes value 1 if the TS is an anomaly and 0 otherwise
meanTot	Average of all data points of the TS to which the stretch belongs
medianTot	Median of all data points of the TS to which the stretch belongs
meanAfter	Average of all data points from the beginning of the TS to which the stretch belongs until the end of the stretch
medianAfter	Median of all data points from the beginning of the TS to which the stretch belongs until the end of the stretch
meanAfterwind	Average of data points from the last 5*window size data points before the beginning of the stretch until the end of the stretch
medianAfterwind	Median of the data points from the last 5*window size data points before the beginning of the stretch until the end of the stretch
sdBefore	Standard deviation of all data points from the beginning of the TS to which the stretch belongs until the beginning of the considered stretch
sdWind	Standard deviation of the data points from the last 5*window size data points before the beginning of the stretch until the beginning of the considered stretch
mean_medianTot	Median of the average values of all stretches of the TS to which the stretch belongs
mean_medianBef	Median of the average values of all stretches previous to the stretch considered
mean-medianWind	Median of the average values of the 10 or 5 previous stretches if overlap is used or not respectively
mean_maxTot	Maximum of the average values of all stretches of the TS to which the stretch belongs
sd_meanTot	Average of the standard deviation values of all stretches of the TS to which the stretch belongs
sd_maxTot	Maximum of the standard deviation values of all stretches of the TS to which the stretch belongs
skew_meanTot	Average of the skewness values of all stretches of the TS to which the stretch belongs
meanLoc	Average of the data points of the considered stretch
medianLoc	Median of the data points of the considered stretch
sdLoc	Standard Deviation of the data points of the considered stretch
amplLoc	Amplitude of the data points of the considered stretch
maxLoc	Maximum of the data points of the considered stretch
skewLoc	Skewness of the data points of the considered stretch

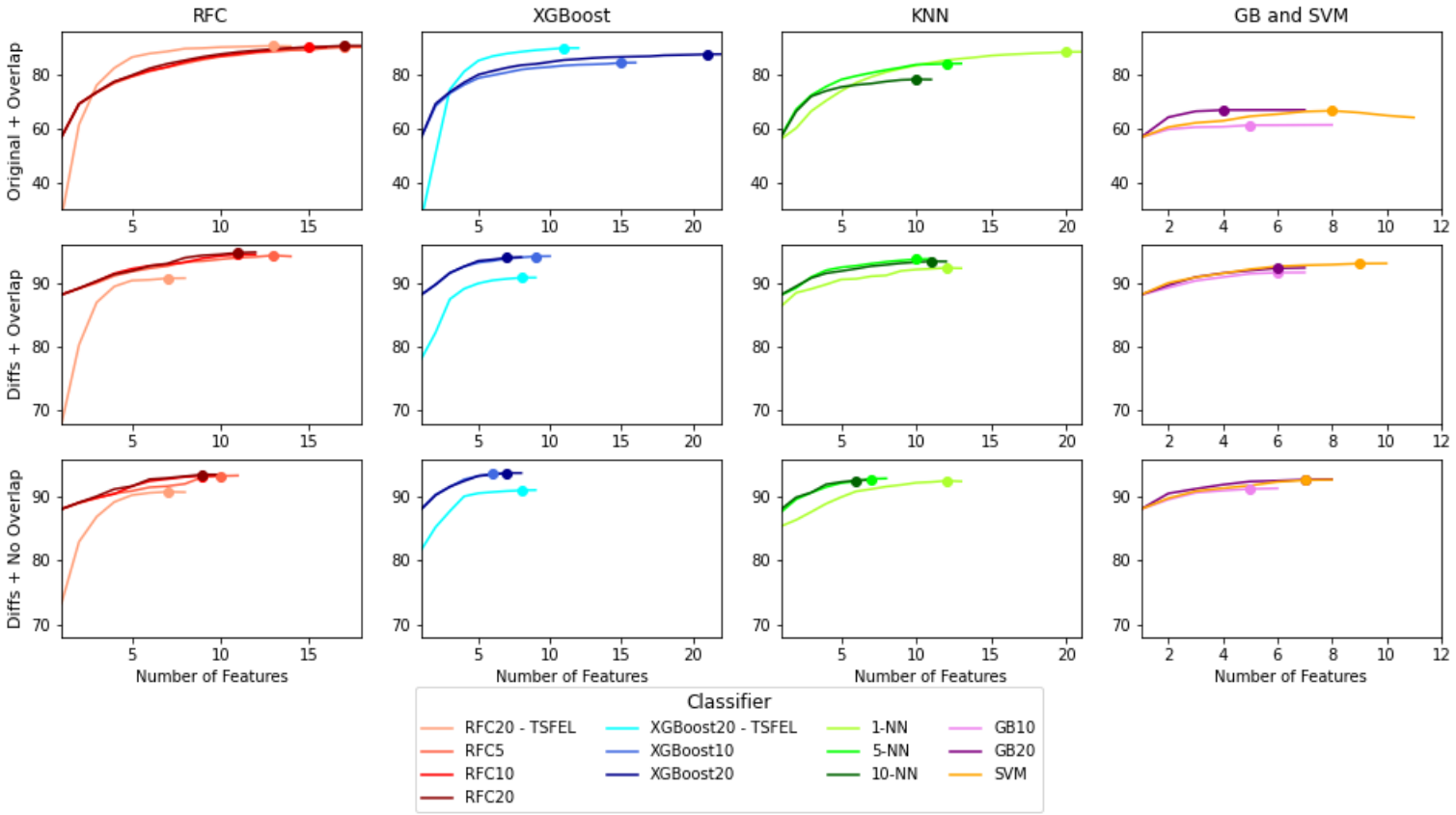

FIGURE A.3: Classifiers: Performance across number of features

It is presented the evolution of the F1-Score with the increase of the number of features across different classifiers. The dots are plotted in the intersection of the number of features and F1-Score obtained when the SC was activated. For all classifiers, it is also presented the performance for the set with one more feature than the ideal to demonstrate the absence of considerable improvement in the performance. Besides, for the Gradient Boost and SVM classifiers in the Original Values, the sets with 3 extra features are plotted to prove that the bad performance is not the result of an early activation of the SC.