Air Quality Time Series Analysis

Data Mining II final report

Stefano Striani
s.striani@studenti.unipi.it
StudentID: 483658

Saeed Choobani
c.saeed@studenti.unipi.it
StudentID: 585084

Niloofar Yousefi
n.yousefi@studenti.unipi.it
StudentID: 584663

Marco Cozzolino
m.cozzolino5@studenti.unipi.it
StudentID: 582370

ABSTRACT

The final report analyzing the dataset of a gas multi-sensor device deployed on the field in an Italian city. This report contains preprocessing the time series and handling Missing values, Time Series Clustering, Sequential pattern mining, Classification, and Outlier Detection.

1 INTRODUCTION

In the Air Quality dataset [1] hourly responses, averages are recorded along with gas concentrations references from a certified analyzer. Due to defined tasks the feature 'PT08.S1(CO)' is used for Clustering, Sequential pattern mining and Classification tasks and the whole dataset is used for Outlier and Anomaly Detection. The dataset contains 9,357 instances and 15 columns.

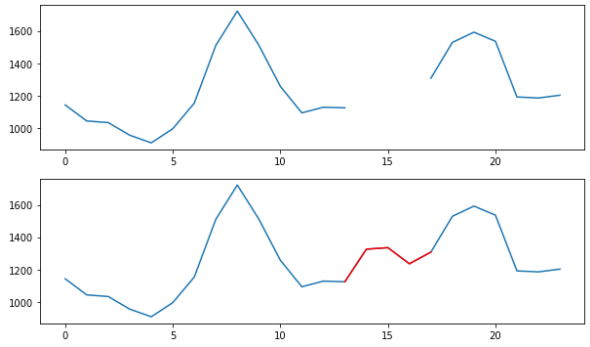
2 MISSING VALUES

Firstly we convert the 'PT08.S1(CO)' column in to Days. The new dataset contains 391 rows as Days and 24 columns as hours. The attribute 'PT08.S1(CO)' of the dataset is not properly composed of missing values, however, there are values that clearly do not reflect truthful observations. Such as all records that have values of -200. these errors can be attributed to different causes. Analyzing the time series related to this attribute we noticed that there are 2 types of missing values in the dataset:

- Full day periods
- Few hours periods

based on this observation we completely eliminated the days that had missing values and to treat with linear interpolation those who had just a few hours with missing records. Fig 2.1 presents a sample Time Series with and with out missing values on the 21st day of the whole days in the dataset.

Fig 2.1 - Handling Missing Values with linear interpolation



We also noticed that all of the time series are not composed of 24 values. the first and the last day of observation consisted respectively of 6 and 15 observations. based on this, we have eliminated these two (First and Last) time series.

3 CLEANING AND PREPROCESSING

Removing four common distortions from the time series is the next step in preprocessing and cleaning the dataset. If we naively try to measure the distance between "raw" time series, we may get very unintuitive results.

The Four steps of Offset Translation, Amplitude Scaling, Smoothing and Trend removal are implemented on the Time Series. Fig 3.1 indicates the distortion removal in each step for a sample number of days in the whole time series. We will use preprocessed time series in clustering, classification and outlier detection tasks.

Fig 3.2 - Sample Number of days in Time Series

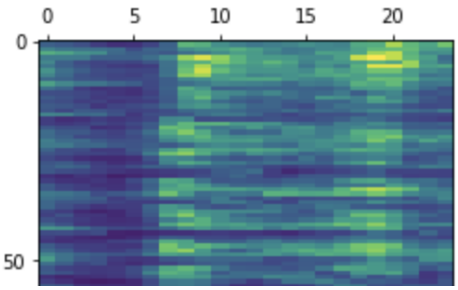
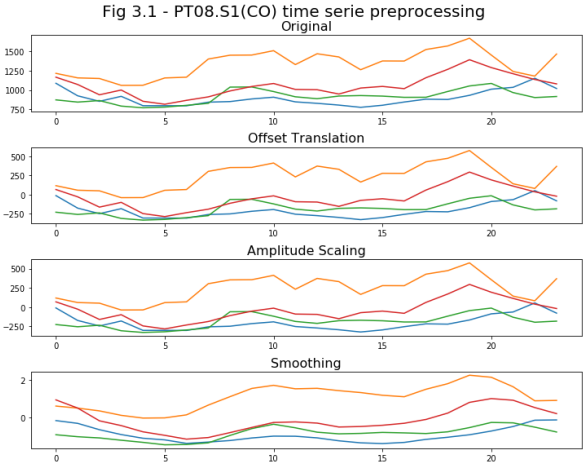


Fig 3.1 - PT08.S1(CO) time serie preprocessing



4 CLUSTERING

After cleaning and modifying the dataset, we have clustered our time series.The algorithm used was performed both using the distance DTW and with Euclidean distance and the results were compared. to create the distance matrix for the DTW we used the external Tslearn library.

4.1 K-MEANS

To estimate the most probable number of clusters for k-means algorithms we created Fig 4.1 . evaluating both distances (DTW and Euclidean), to get a good SSE score without increasing the number of clusters too much, we have chosen the value of **k = 15**. the algorithm is particularly inefficient using DTW and his run time was 3 hours with all the parameters set to minimum (n-init=1, Max-Iter=100). the results obtained are not as good as those obtained with the Euclidean distance.

Fig 4.1 - SSE vs Number of Clusters

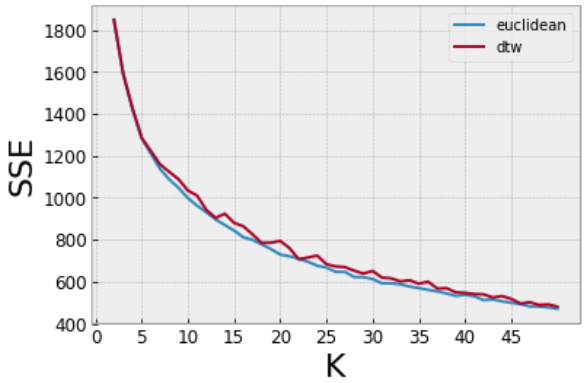


Table 4.1	Silhouette score	Clusters
DTW	0.042	15
Euclidean	0.1144	15

4.2 DBSCAN

In order to perform clustering using DBSCAN we initially had to evaluate the parameters (eps to use for the algorithm.) In Fig 4.2 we can see the density distribution for choosing the eps parameter. It is noted that both curves have a Gaussian distribution however the DTW with respect to the euclidean has more narrow despair of the points.

Fig 4.2 - eps estimation

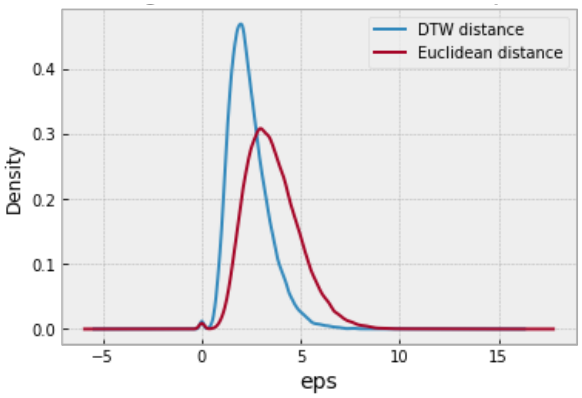


Fig 4.3 shows the variation in the number of clusters obtained with the variation of eps. Also, in this case, there is a clear difference between DTW and Euclidean.

Fig 4.3 - Number of Clusters and Noises vs eps

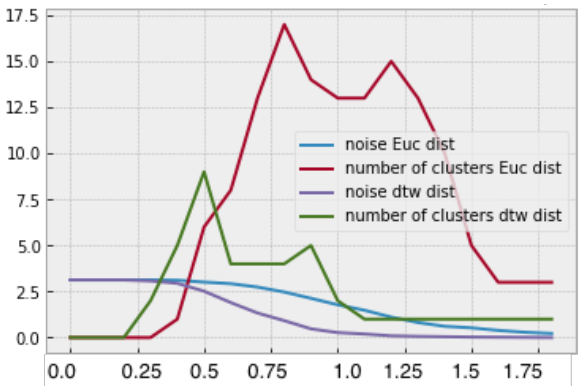


Table 4.2, in addition to the algorithm parameters, are reported the silhouette score and the data relating to the major clusters. From the silhouette score we notice that both distances do not allow to obtain good results, moreover the algorithm classifies most of the time series as noise points without being able to evaluate real clusters.

Table 4.2	**DTW**	**Euclidean**
Epsilon	0.75	1.3
Min sample	2	3
Silhouette	-0.2447	-0.01723
N. of clusters	7	15
Biggest Cluster	148	185
Noise point	170	120

4.3 HAC

Hierarchical clustering seems to be the most promising of the found algorithms. moreover, it is not computationally more expensive than the Euclidean distance version. Table 4.3 shows the number of clusters obtained for the two algorithms. Considering Fig 4.3 performing a 'cut' at distance 6 for the DTW we get 3 clusters and a record is classified as an outlier while using Euclidean distance and performing a 'cut' at distance 7 we get 4 clusters with 1 outlier. in both cases, the clusters obtained seem to be well separated, which means that the time series within the clusters differ significantly.

Fig 4.3 - HAC dendrogram DTW distance

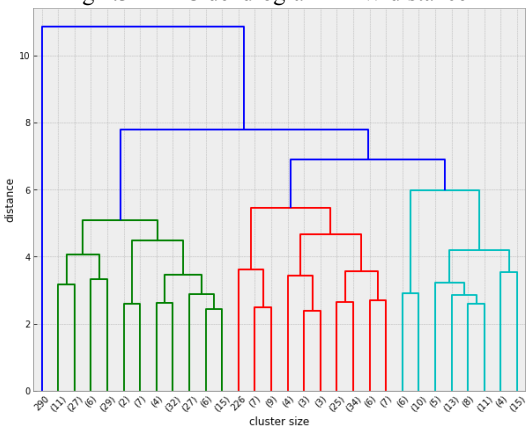


Fig 4.3 - HAC dendrogram EUC distance

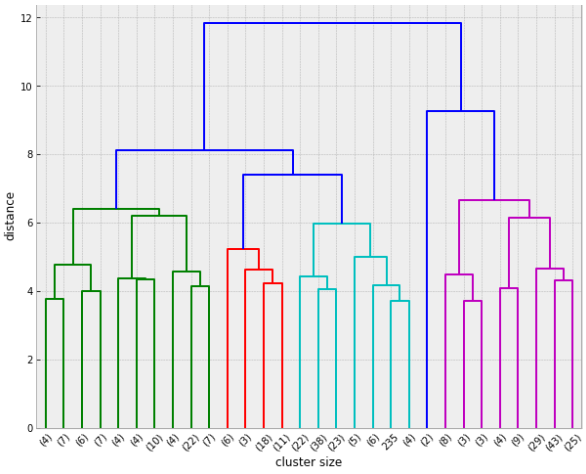
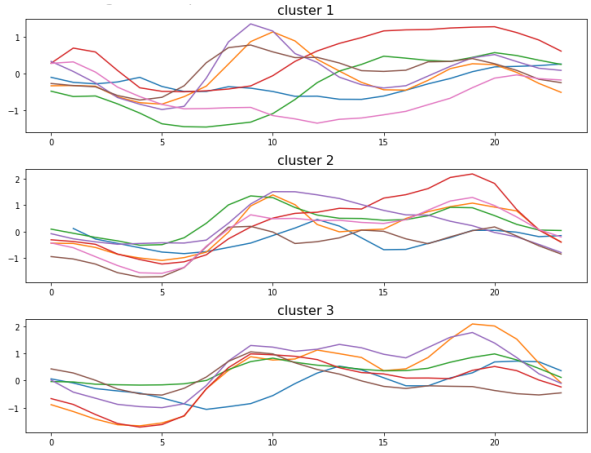


Table 4.3	Cut	Clusters
DTW	6	3 (1 Outlier)
Euclidean	7	4 (1outlier)

Considering the different approaches used to perform the clusters, we believe that k-means is not suitable for our dataset. Moreover, using DTW, it turns out to be very time-consuming (4 hours for estimating the best number of clusters)DBSCAN does not suffer from high complexity but the results obtained, both with DTW and Euclidean, are not significant. For our purposes, we believe that the use of hierarchical clustering is more advantageous and manages to better understand the differences between different time series. Fig 4.4 (below) are examples of time-series present in the 3 different clusters obtained through HAC with DTW distance.



5 SEQUENTIAL PATTERNS

Before implementing any sequential pattern mining algorithm on the dataset we must

somehow discretize the data. The first and simplest discretization method would be changing each column's data into ranges of data in different bids. The value of the columns is in the range of 600 to 2200, so we'll create labels for the bids of 200 in each data column and change their datatypes into strings.

For the implementation method, we're using SPMF software and SPAM algorithm. SPAM is an algorithm for discovering frequent sequential patterns in a sequence database. The input of SPAM is a sequence database and a user-specified threshold. SPMF needs a specific input file format. It is a text file where each line represents a sequence from a sequence database. The value "-1" indicates the end of an itemset. The value "-2" indicates the end of a sequence (it appears at the end of each line). The goal of the task is to discover sequential patterns of at least length 4 which are contiguous, this means that the Max-Gap between them is 1.

With the support of 45%, we can reach the pattern in Table 5.1. This pattern is the same in 176 rows of the data.

Table 5.1 - Frequent pattern

8003 -1	8004 -1	8005 -1	8006 -1

Due to this pattern nearly in the half of the year hourly averaged concentrations for CO from 3 to 6 A.M are in the range of 800-1000.

6 CLASSIFICATION

For the classification task, we need to add a class label which is True for weekends days and False for week days. In this report, we're implementing K-NN and Kernel SVM classification methods with various parameters and comparing the results.

6.1 K-NN CLASSIFIER

In order to obtain the best number of neighbors for the K-NN algorithm, we fit the model various times using cross-validation. Fig 6.1 Indicates different measurements for Euclidean and DTW in a range of different numbers for the neighbors. The plot shows that the best number of nearest neighbors for DTW is 6 and for Euclidean is 3 according to the accuracy. It's obvious that Euclidean is providing us with better results.

Fig 6.1

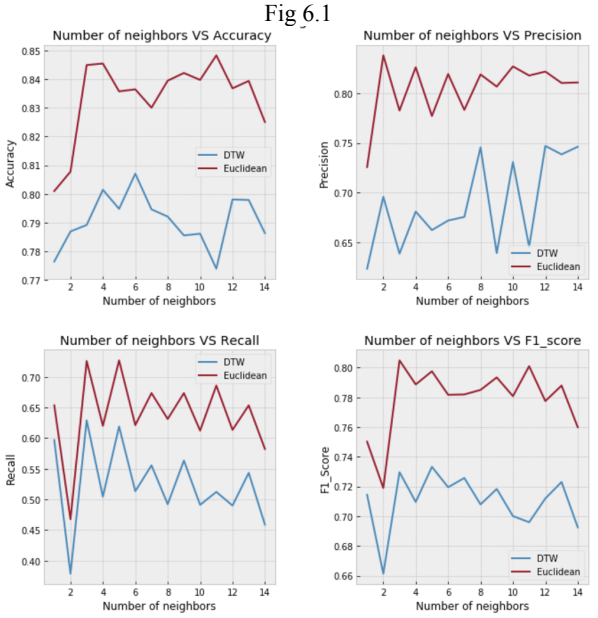


Table 6.1	Accuracy	F1-Score	Recall	Precision
DTW	0.8070	0.7195	0.5133	0.6720
Euclidean	0.8450	0.8048	0.7256	0.7826

6.2 SVM CLASSIFIER

In the search for finding a better classifier, we're testing the Kernel SVM on the dataset. GridSearchCV functions suggest the best input parameters. (Table 6.2)

Table 6.2

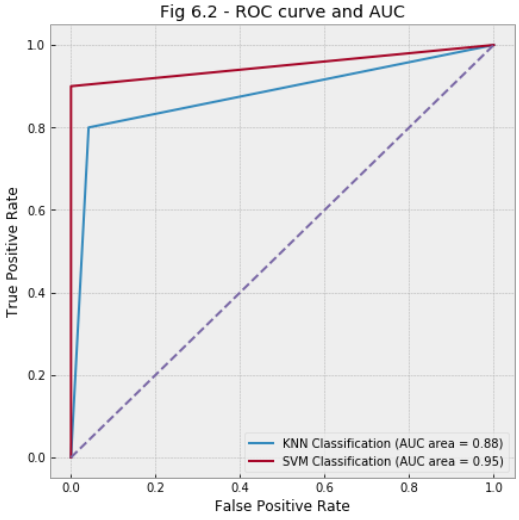
Kernel	Degree	C
Rbf	3	1.0

The best results using cross validation are reported in the Table 6.3.

Table 6.3	Accuracy	F1-Score	Recall	Precision
SVM	0.8635	0.8232	0.6944	0.8240

We implement the ROC curve by plotting True positive rate against False positive rate to evaluate the performance of our two best classifiers. As indicated in Fig 6.2 the SVM classifier provides us with a better result. The AUC (area under the cure) for SVM model is bigger than K-NN model.

Fig 6.2 - ROC curve and AUC



7 OUTLIER AND ANOMALY DETECTION

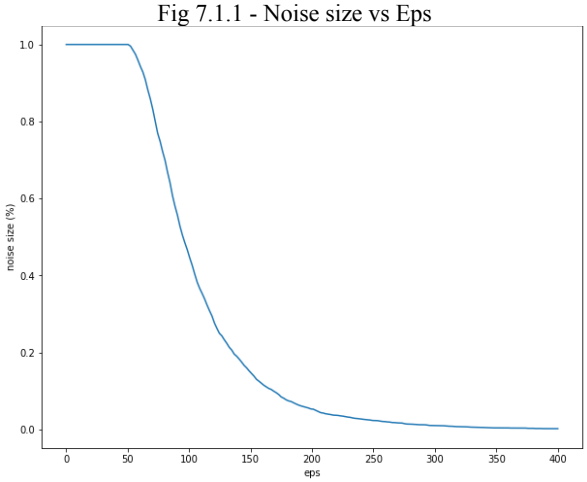
The task objective is to identify the top 1% outliers of the whole dataset. As we've mentioned before all the columns contain missing values. On top, the column 'NMHC(GT)' contains missing values in more than 90% of the instances. We remove this column totally and then remove the rows with any missing value from the whole dataset. The final dataset got 6941 rows of data. All distortion removal methods (Offset Translation, Amplitude Scaling, Smoothing and Trend removal) are also implemented column by column on the whole dataset.

7.1 DISTANCE BASED

The first method we're examining for determining outliers is the one based on the distance between the points of a dataset. once a circle of radius eps and a percentage pi is determined, a point is considered an outlier if there are less than pi % of points within the circle.

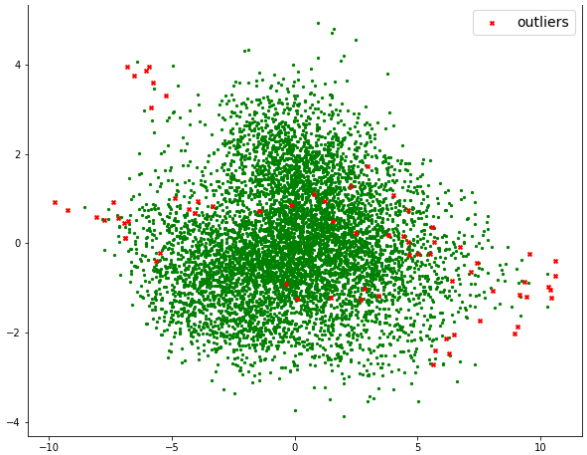
In order to implement this outliers detection method, we used some of the features taken from the clustering algorithm DBSCAN. through this algorithm, we were able to divide the noise points from the others based on their mutual distance.from the curve, in Fig 7.1.1 we determined the best value of esp (around 290) to determine 1% of the most probable outliers.

Fig 7.1.1 - Noise size vs Eps



In Fig 7.1.2 we observe a 2-d representation of the dataset obtained by PCA method. Most of the detected outlier nodes are in the outer margin of the data set.

Fig 7.1.2

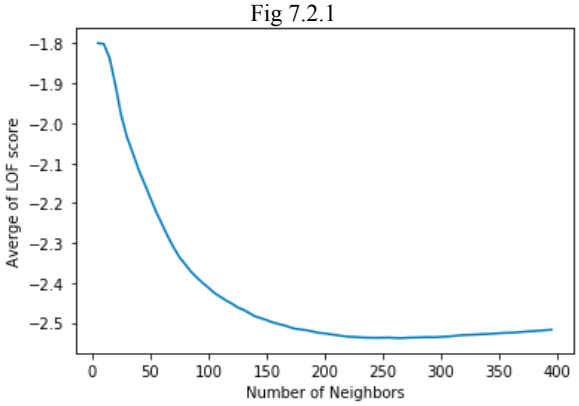


7.2 DENSITY BASED (LOF)

The Local Outlier Factor (LOF) algorithm is an unsupervised anomaly detection method which computes the local density deviation of a given data point with respect to its neighbors. In order to find the best number of neighbors as the input parameter, we measure the average score of top

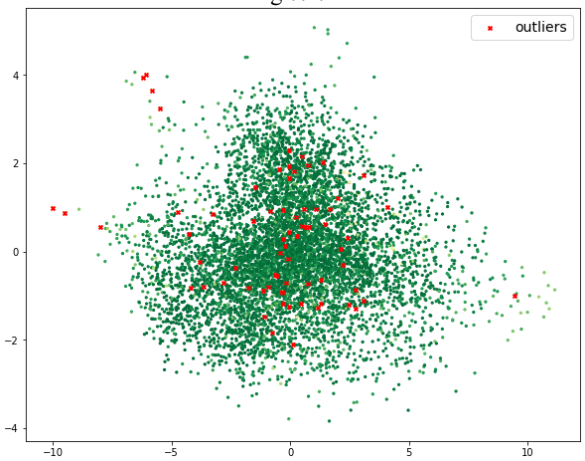
70 points. (1% of whole dataset). Fig 7.2.1 shows the variation of this measurement during the changes in the number of neighbors. It seems like 150 would be the best number, since afterwards the average plot begins increasing.

Fig 7.2.1



In order to visualize the outliers we perform PCA method and decrease all the dimensions in the dataset into 2 components. The green gradient color on each point indicates their LOF score and red data points are the outliers. Fig 7.2.2 visualizes the LOF anomaly detection.

Fig 7.2.2



7.3 LIMIT CHECKING

Statistical methods for outlier detection is an alternative approach to fault detection based on limit checking with constant or linear thresholds. In this case we are determining the values more than **Average ± 5*Std** as the outliers in each column. With this method we will have **161** instances as outliers.

7.4 ISOLATION FOREST

IsolationForest is an ensemble regressor. IF builds an ensemble of random trees for a given data set and anomalies are the points with the shortest average path lengths. The IsolationForest 'isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

IF algorithm is almost robust to the input parameters [2]. We will use the parameters in Table 7.4 for implementation.

Table 7.4 - IF parameters

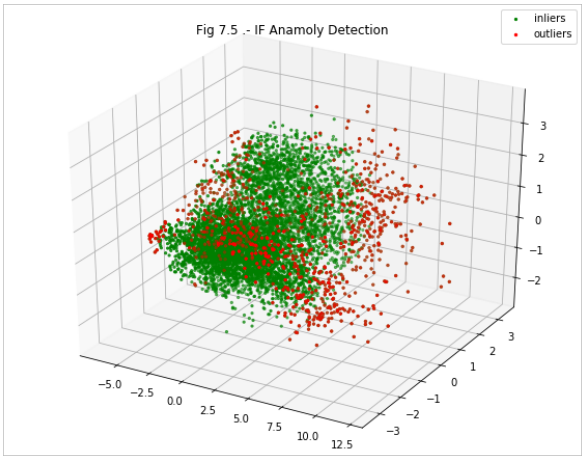
n_estimators	max samples	max features	contamination
100	'auto'	1.0	float(.12)

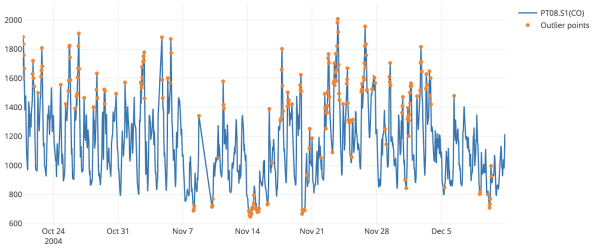
Table 7.5 reports the Final results of IF on the dataset.

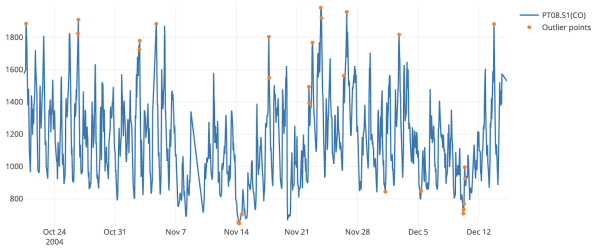
Table 7.5 - IF results

Score Average	Outliers	Inliers
0.05866	833	6108

In order to visualize the results we compress all dimensions into 3 components using PCA method. Fig 7.5 illustrates a portion of PT08.S1(CO) time series and the Outlier points. And also represents a more general visualization of the outliers in all dimensions using PCA method.





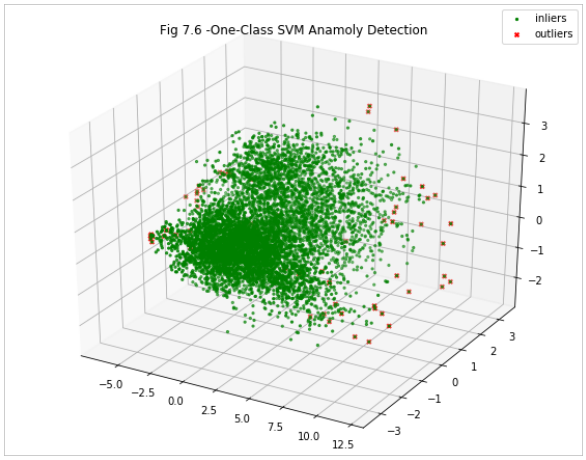


7.5 ONE_CLASS SVM

The One-Class SVM is known to be sensitive to outliers and thus does not perform very well for outlier detection. This estimator is best suited for novelty detection when the training set is not contaminated by outliers. That said, outlier detection in high-dimension, or without any assumptions on the distribution of the inlying data is very challenging, and a One-class SVM might give useful results in these situations depending on the value of its hyper-parameters. Table 7.7 presents the hyper-parameters of the One-Class SVM model implementation on the dataset. Obviously the model will detect 69 outliers and 6245 inliers. Fig 7.6 illustrates a portion of PT08.S1(CO) time series and the Outlier points. And also represents a more general visualization of the outliers in all dimensions using PCA method.

Table 7.6 - One-class SVM hyper-parameters

outliers fraction	kernel	gamma
0.01	Rbf	0.01



8 CONCLUSIONS

In this report we analyzed Clustering, Sequential pattern mining, Classification, and Outlier Detection of a gas multi-sensor device deployed on the field in an Italian city. The analysis came up with 3 main clusters in the dataset, found 176 rows of the dataset with the same pattern in sequential pattern mining, reached 0.8635 accuracy in classification and examined 6 different methods of anomaly detection on the data set. Since the anomaly detection tasks are unsupervised we can't determine which method is doing better.

REFERENCES

[1] https://data.world/uci/air-quality
[2] Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou Isolation Forest