Anomaly Detection in Time Series: A Comprehensive Evaluation

Stanislas du Ché stanislasduche@gmail.com
Théo Di Piazza theo.dipiazza@gmail.com

March 27, 2023

1 Introduction and Contributions

"Anomaly Detection in Time Series: A Comprehensive Evaluation" presents a very interesting benchmark and comparison of different algorithms. The anomaly detection litterature is very dense and many different methods are proposed. This article aims at first to cluster and categorize the algorithms and then compare their performances and multiple types of outliers and datasets.

This evaluation covers the areas of Classic Machine Learning, Deep Learning, Signal Analysis, Data Mining, Stochastic Learning, Statistics and Outlier Detection. To compare the algorithms, the paper first spots the different types of outliers that algorithms will be tested on : amplitude, extremum, frequency, mean, pattern, pattern shift, platform, trend and variance. The paper selects a wide range of datasets to be tested and creates fake datasets containing special types of outliers thanks to GutenTAG software.

Moreover, the paper clusters the algorithms in the following methods : Forecasting, Reconstruction, Distance, Encoding, Distribution and Tree Methods. What would be interesting to understand is how the different methods adapt to each type of outlier. As some algorithms may be very efficient to detect a type of outlier, it may struggle to identify an other type. This rely on the understanding of an outlier and its mathematical derivation.

Our work will be to select a few methods and explain their mathematical interpretation of an outlier. Then, we will study their implementation and their ability to spot "adequate" or "unexpected" outliers and explain some struggles may occur.

The code we used the algorithm implementation provided by the paper and relied on GutenTAG to created datasets containing outliers we wanted to test.

2 Method

During the project, we selected a few algorithm from the different anomaly detection methods. Our goal is to understand how each method is representing an anomaly and how each method is performing on differents datasets.

We will focus on 3 methods : K-Means (distance method), Normalizing Flow (distribution method) and SARIMA (forecasting method).

2.1 KMeans description

K-Means [2] is a widely used unsupervised clustering algorithm able of handling multivariate data. It can be used in the case of anomaly detection for a time series. According to the paper, it is considered an unsupervised classic ML algorithm belonging to the family of distance-based methods.

Pre-processing The first step in K-Means is to decompose the time series into a finite number of fixed-size subseries (*window size*), using a sliding window that moves by *stride* units for each sample. Each window (input of K-Means) is a vector of length *window size* representing a part of the time serie.

Method Then in the classic way, the objective is to partition the observations into k sets $S = \{S_1, S_2, ..., S_k\}$ to minimize the within-cluster sum of squares. The quantity to minimize is :

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2$$

Where x is a window (subserie of length *window size*) and μ_i the centroid of S_i. Once this step is completed, each subset is associated with a cluster.

Compute score The final score of a point of the initial time series is the average of the distances (Frobenius norm) between the subseries to which this point belongs and the centroids of the cluster associated with each subserie. Then, scores are normalized to be between 0 and 1.

2.2 SARIMA description

SARIMA [4] is an unsupervised model based on statistical approach, belonging to the family of forecasting-based methods. This method is capable of handling univariate data.

Pre-processing The dataset is separated into 2 sets: the first set is used to train the SARIMA model. The initial train set is composed of *train window size* samples.

Method SARIMA is an improved version of a mix between the AR and the MA method, with 7 parameters : (p, d, q)(P, D, Q, s) where p, q, P, Q indicates how many periods we go back for AR and MA processes; d the degree of differencing and D the degree of seasonal differencing.

Then the model makes a prediction for the whole data set and at each iteration the model is updated with the predictions made over a window of size *forecast window size*. When all the points have been processed, the scores can be calculated.

Compute score For a given point, the score is simply the absolute value of the difference between the prediction made by the SARIMA model and the observation. The scores are then normalised to be between 0 and 1. Intuitively, a point is considered an anomaly if its observed value is very different from the value predicted by the model.

2.3 Normalizing Flows description

Normalizing Flows [3] (NF) is a supervised learning algorithm based on a neural network architecture, belonging to the family of distribution-based methods. This method is capable of handling multivariate data.

Pre-processing The first step in NF is to decompose the time series into a finite number of fixed-size subseries, using a sliding window that moves by *stride* units for each sample. In multivariate case, a sample corresponds to the concatenation of windows of length *window size* extracted from each dimension. Hence, each input of NF is a vector of length *window size* * *number of dimension*. A window is considered as an anomaly if at least one sample from the window is an anomaly.

Method NF consists of 2 steps: the first step is to train a model to transform a known distribution into a more complex distribution that matches the data. It is done by applying a series of invertible transformations to the known distribution, each of which is referred to as a "flow". Once this model is trained, it can be used to generate data from the learned distribution. The second step is to train a second binary classification model to predict whether a sample is normal or anomalous with respect to the learned distribution. For a window, its anomaly score corresponds to the prediction of the classifier.

Compute score NF outputs anomaly scores for windows. The results require post-processing : the scores for each point of the initial time serie can be assigned by aggregating the anomaly scores for each window the point is included in.

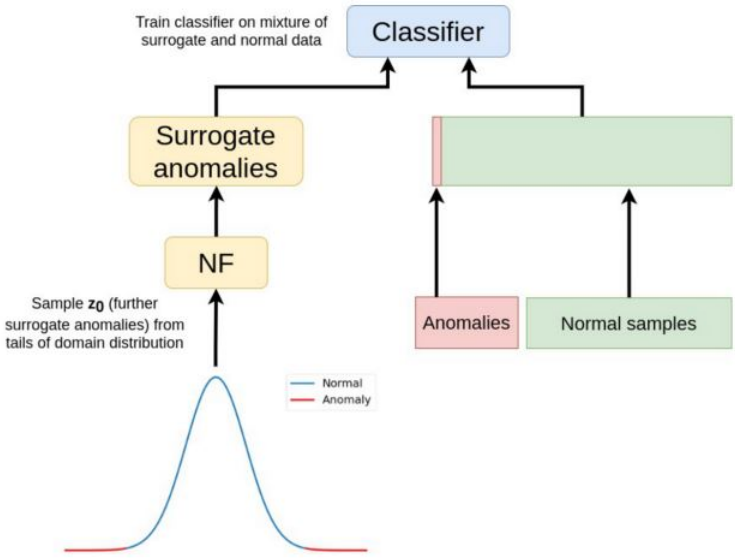


Figure 1: Diagram of NF fow anomaly detection. Source : [??]

3 Data

The objective of our paper review is to understand how methods describe and encode anomalies; that's why we don't want to work on complex data. In a first step, we will focus on anomalies from the GutenTAG dataset, and then work on more complex anomalies found on the internet via the real "NAB" datasets.

4 signals from GutenTAG, with different properties, will be used :
A1. sinus-noise-10% a signal having the shape of a sinus with noise, with 1 anomaly and without trend ; **A2. poly-same-count-2** a polynomial signal with 2 anomalies, with an increasing then decreasing trend ; **A3. cbf-trend-linear** a signal corresponding to the Cerebral Blood Flow with 1 anomaly ; **A4. ecg-diff-count-7** a signal obtained by Electrocardiography with 7 anomalies.

In a second step, we will use real signals found on the internet [5].

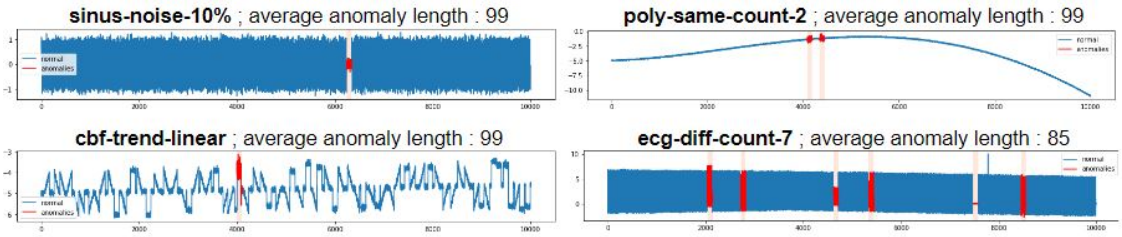


Figure 2: A. Tests set of used datasets from GutenTAG.

B1. Artificial Daily Flat Middle an artificial dataset with 1 anomaly associated with a constant value ; **B2. temperature-system-fail** ambient temperature in an office setting with 2 anomalies related to system failure ; **B3. nyc-taxi** number of NYC taxi passengers, with 5 anomalies for the NYC marathon, Thanksgiving, Christmas, New Years day, and a snow storm ; **B4. twitter-volume-fb** number of mentions on Twitter for @Facebook every 5 minutes with 2 anomalies.

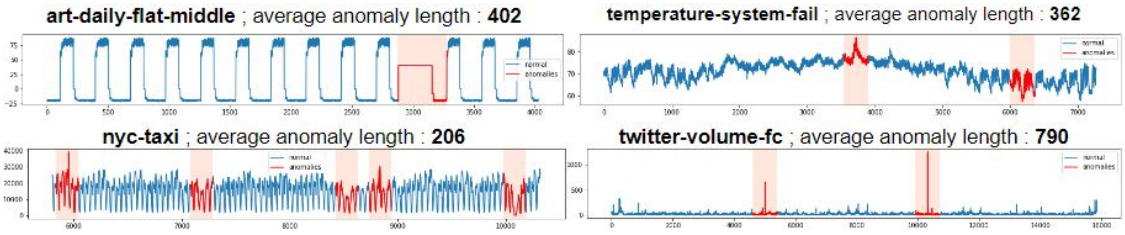


Figure 3: B. Own datasets from NAB.

4 Results

Results on GutenTAG datasets, please see **Appendix 2, Table 1** for detailed metrics.

GutenTAG : sinus-noise type signals (Appendix 3) : The particularity of poly is the smooth change of trend. The 3 methods used with the right hyper-parameters have no difficulty in identifying anomalies. For SARIMA, the method is particularly slow with a small prediction window.

GutenTAG : poly type signals (Appendix 4) : The particularity of poly is the smooth change of trend. The 3 methods used with the right hyper-parameters have no difficulty in identifying anomalies.

GutenTAG : CBF type signals (Appendix 5) : The particularity of CBF is that it presents strong disturbances, without trends or repeating patterns. SARIMA and K-Means seem to be inefficient and difficult to optimise on this type of data because they are very sensitive to the slightest disturbance. NF has a slightly easier time detecting the anomaly thanks to the training phase.

GutenTAG : ECG type signals (Appendix 6) : The particularity of ECGs is the repetition of the same pattern. K-Means seems efficient on this type of signals because it is sensitive to the slightest variation of the signal pattern. For NF, very efficient because the slightest change in amplitude is considered as an anomaly. For SARIMA, the prediction error takes over and prevents the detection of the anomaly.

Results on datasets from internet, please see **Appendix 2, Table 2** for detailed metrics and **Appendix 7** for visualization. Looking at the metrics obtained, we see that the results are significantly worse than on the GutenTAG dataset, which is consistent with the fact that the data is real

and therefore has more noise. For the 4 datasets, it was not possible to converge NF because of a gradient explosion. From a qualitative point of view, K-Means is globally efficient because when the window is well chosen, the anomaly is detected. As for SARIMA, it is more difficult to obtain satisfactory results: when the model fails to model the series correctly, the prediction error is the same everywhere, and it is not possible to detect the anomaly.

Finally, some comments on the influence of the parameters of the methods used :

K-Means : Influence of number of neighbours (Appendix 8): This method does not seem to be suitable for signals with a trend or disturbance. For K-Means algorithm, robustness can be obtained by increasing the k but requires more computational resources and leads to over-fitting. Moreover, having an appreciation of the size of the anomalies and the potential behaviour of the signal helps tuning the hyper-parameters.

SARIMA - Influence of window prediction (Appendix 9) : At the first iteration, the model is trained on the part of the time serie (poly-count-diff-2) with an increasing trend. On anomaly detection, the trend becomes decreasing. Thus, when the prediction window is too small, the prediction error propagates more and more, until it is no longer able to identify the anomaly. For SARIMA, an anomaly is a set of points whose behaviour does not correspond to the results expected by the model.

5 Conclusion

This project was the opportunity to explore 3 anomaly detection methods belonging to different families. NF, a supervised algorithm, is able to recognise anomalies if they have already been seen during the training phase. K-Means is efficient in detecting anomalies that present amplitude variations with its neighbourhood. Finally, SARIMA identifies an anomaly as a set of points that differs from the series model, but is very sensitive to the quality of this model.

References

[1] **Anomaly Detection in Time Series: A Comprehensive Evaluation** - S. Schmidl, P. Wenig, T. Papenbrock (2022)

[2] **K-Means: Fault detection by mining association rules from house-keeping** - T. Yairi, Y. Kato, and K. Hori (2001)

[3] **NFAD: Fixing anomaly detection using normalizing flows** - A. Ryzhikov, M. Borisyak, A. Ustyuzhanin, D. Derkach (2021)

[4] **SARIMA : Comparing Prediction Methods in Anomaly Detection: An Industrial Evaluation** - R. Greis, T. Ries, and D. Cu Nguyen (2018)

[5] **NAB dataset : Unsupervised real-time anomaly detection for streaming data** - S. Ahmad,A. Lavin, S. Purdy, Z. Agha (2017)

Appendix 2 : Results of K-Means, SARIMA and NF on datasets from GutenTAG.

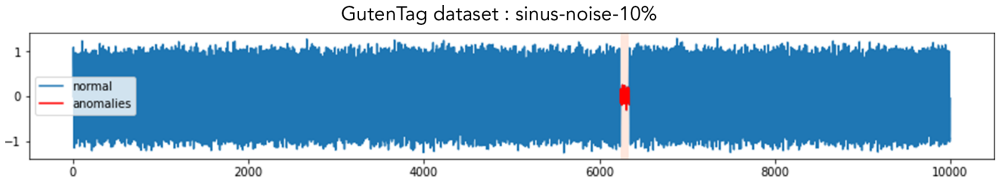
Table 1: Results of K-Means, SARIMA and NF on GutenTAG.

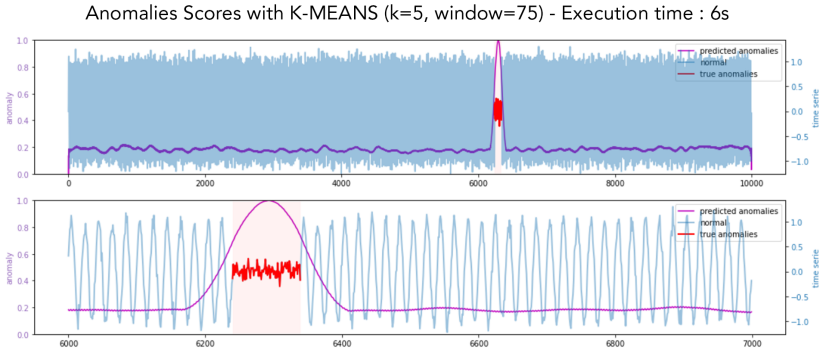
Dataset	**Method**	**AUC-ROC**	**AUC-PR**	**AUC-P$_T$R$_T$**	**Time**
sinus-noise-10%	K-Means	0.99	0.99	0.87	2s
sinus-noise-10%	SARIMA	0.98	0.85	0.54	161s
sinus-noise-10%	**NF**	**1.0**	1.0	0.76	35s
poly-same-count-2	**K-Means**	**0.99**	0.99	0.98	9s
poly-same-count-2	SARIMA	0.75	0.40	0.52	292s
poly-same-count-2	NF	0.99	0.93	0.66	45s
cbf-trend-linear	**K-Means**	**0.99**	0.83	0.78	13s
cbf-trend-linear	SARIMA	0.96	0.47	0.53	84s
cbf-trend-linear	NF	0.94	0.23	0.54	49s
ecg-diff-count-7	**K-Means**	**0.96**	0.50	0.57	11s
ecg-diff-count-7	SARIMA	0.54	0.07	0.23	188s
ecg-diff-count-7	NF	0.82	0.60	0.58	42s

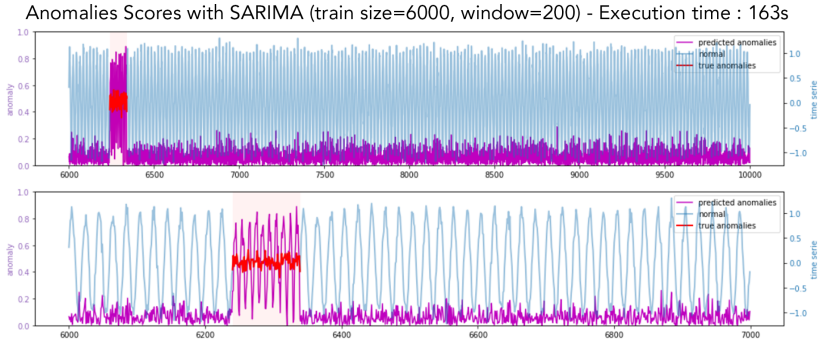
Table 2: Results of K-Means, SARIMA and NF on GutenTAG. GE for Gradient Explosion.

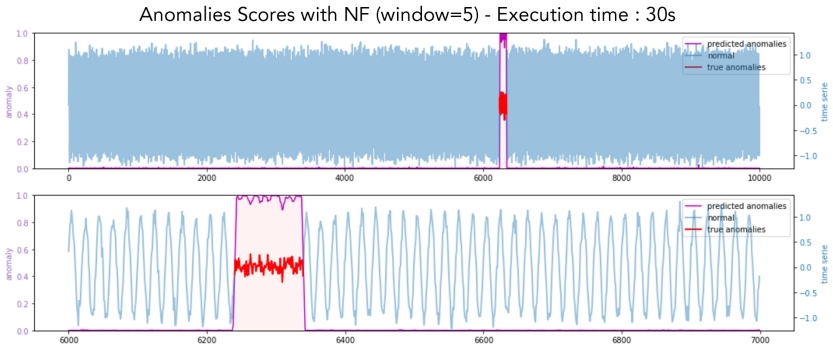
Dataset	**Method**	**AUC-ROC**	**AUC-PR**	**AUC-P$_T$R$_T$**	**Time**
art-daily-flatmiddle	K-Means	0.70	0.35	0.61	10s
art-daily-flatmiddle	**SARIMA**	**0.75**	0.17	0.08	11s
art-daily-flatmiddle	NF	GE	GE	GE	GE
temperature-system-failure	K-Means	0.66	0.33	0.47	1s
temperature-system-failure	**SARIMA**	**0.80**	0.27	0.22	66s
temperature-system-failure	NF	GE	GE	GE	GE
nyc-taxi	**K-Means**	**0.91**	0.76	0.76	10s
nyc-taxi	SARIMA	0.80	0.23	0.42	160s
nyc-taxi	NF	GE	GE	GE	GE
twitter-volume-FB	K-Means	0.57	0.23	0.46	11s
twitter-volume-FB	**SARIMA**	**0.68**	0.17	0.44	63s
twitter-volume-FB	NF	GE	GE	GE	GE

Appendix 3 : Results of K-Means, SARIMA and NF on sinus-noise10% from GutenTAG.

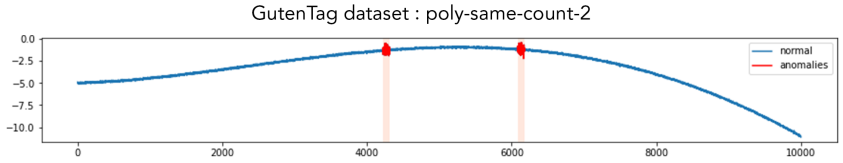


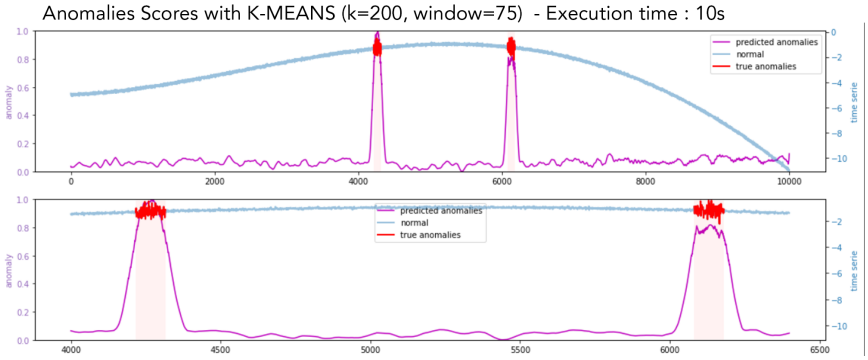


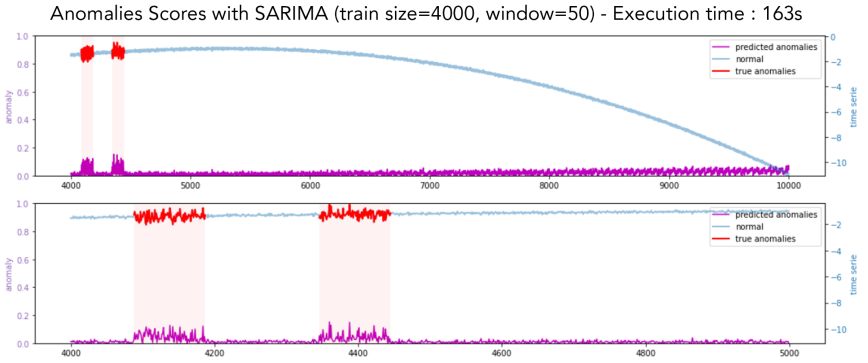


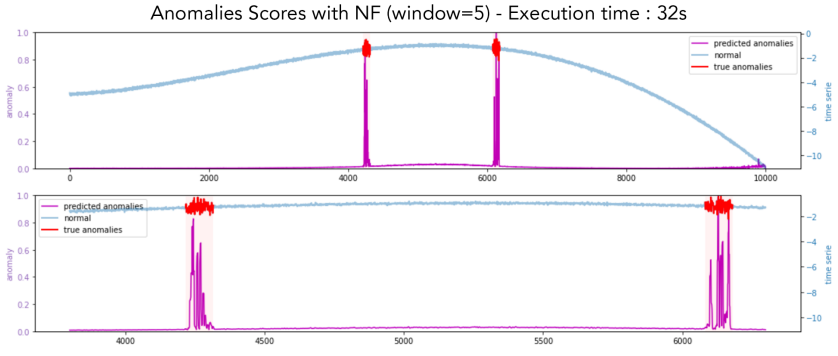


Appendix 4 : Results of K-Means, SARIMA and NF on poly-same-count-2 from GutenTAG.

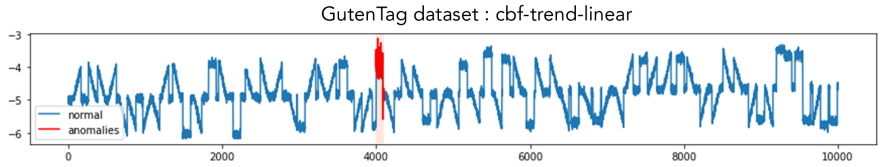


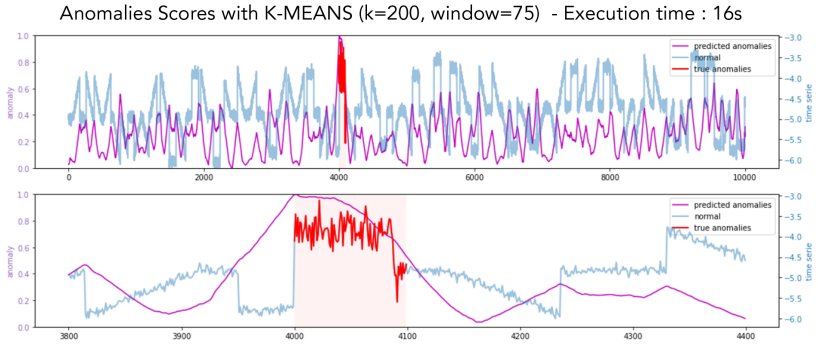


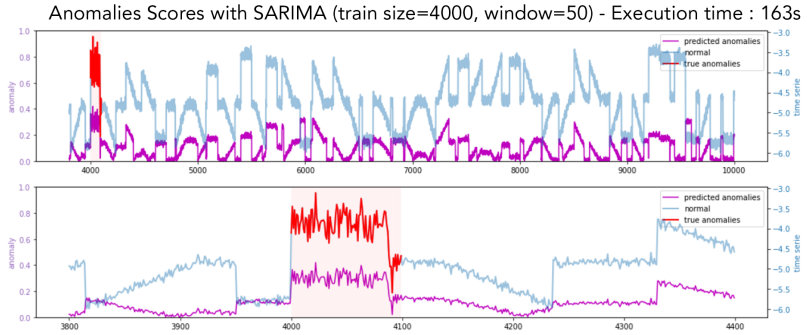




Appendix 5 : Results of K-Means, SARIMA and NF on cbf-trend-linear from GutenTAG.

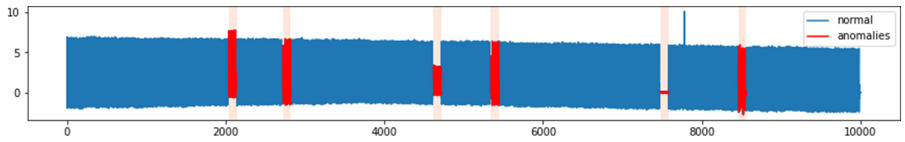




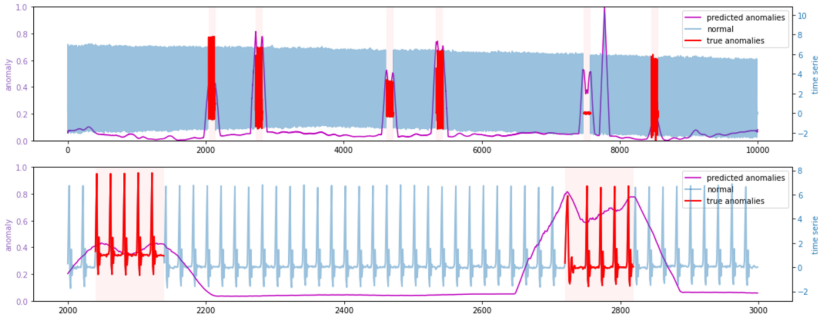


Appendix 6 : Results of K-Means, SARIMA and NF on ecg-diff-count-7 from GutenTAG.

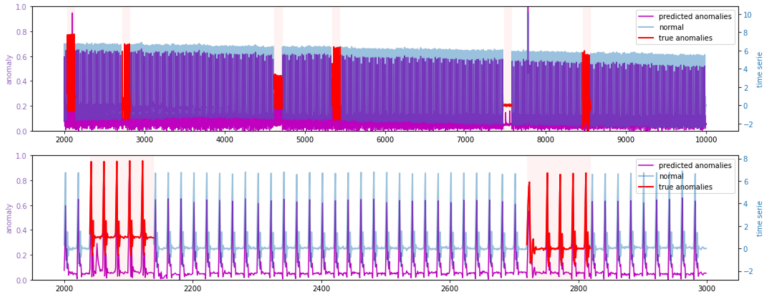
GutenTag dataset : ecg-diff-count-7



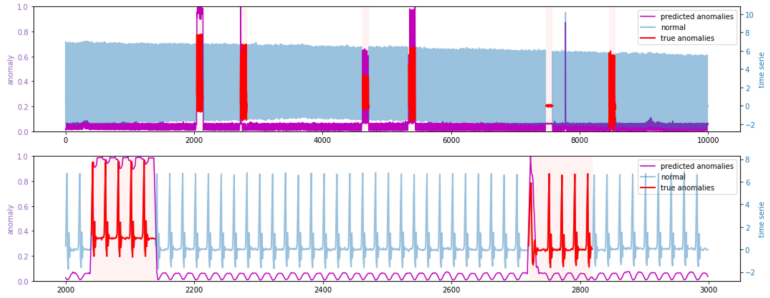
Anomalies Scores with K-MEANS (k=200, window=75) - Execution time : 11s



Anomalies Scores with SARIMA (train size=2000, window=50) - Execution time : 163s

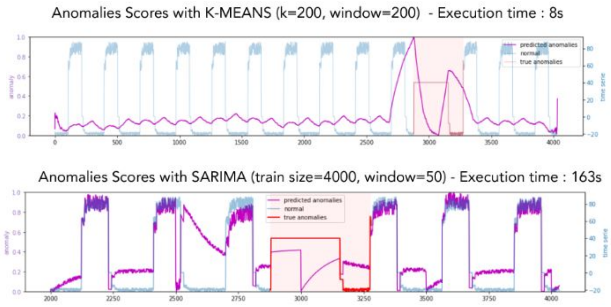


Anomalies Scores with NF (window=5) - Execution time : 38s

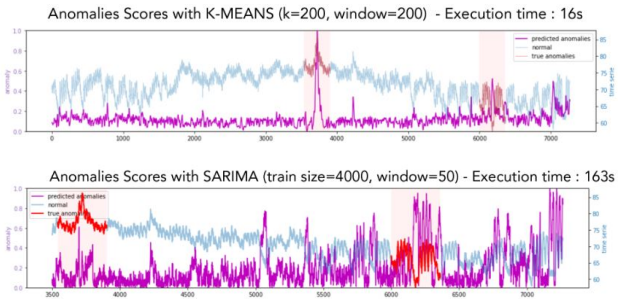


Appendix 7 : Results of K-Means, SARIMA and NF on NAB datasets.

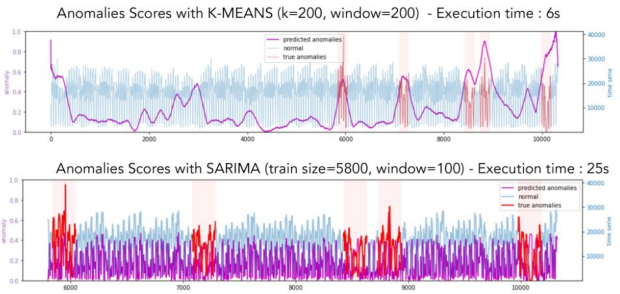
Results on art-daily-flatmiddle



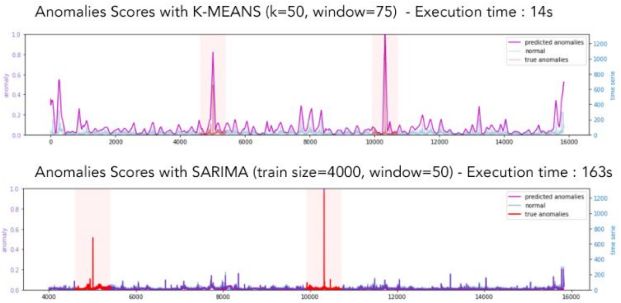
Results on temperature-system-failure



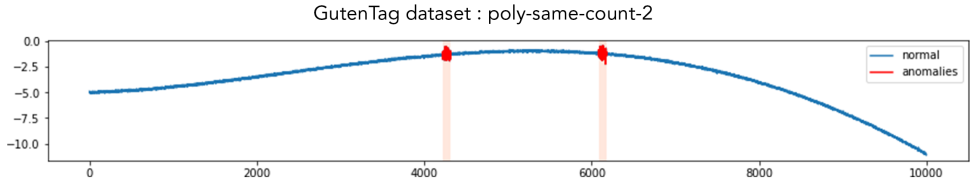
Results on nyc-taxi

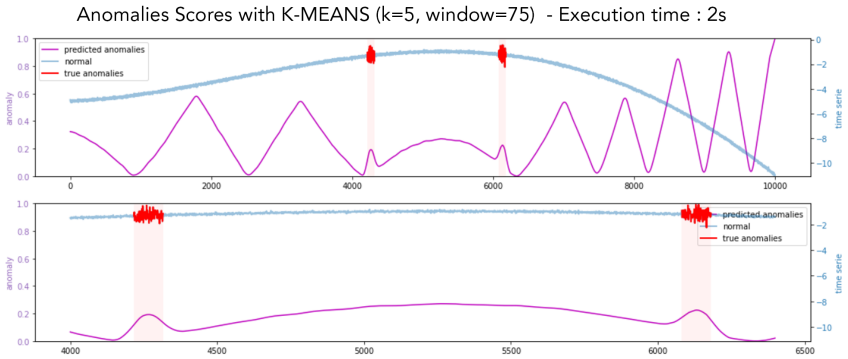


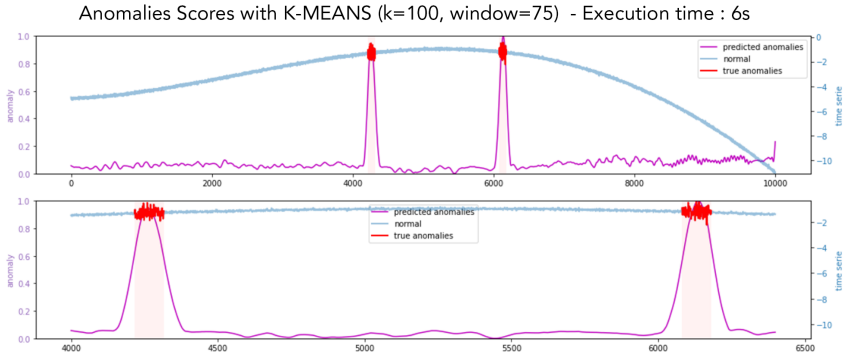
Results on twitter-volume-FB

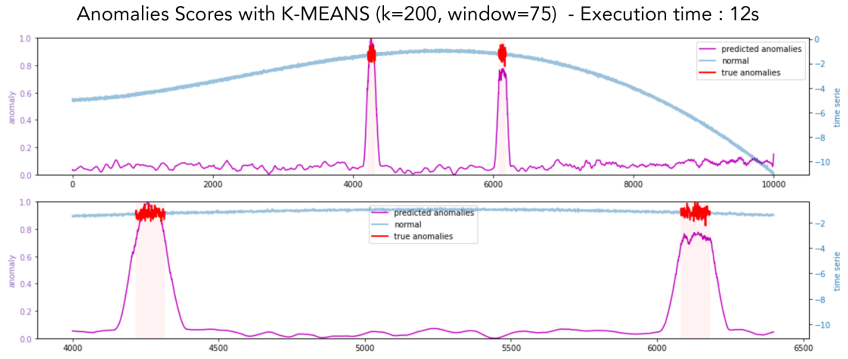


Appendix 8 : Influence of neighbours for K-Means.



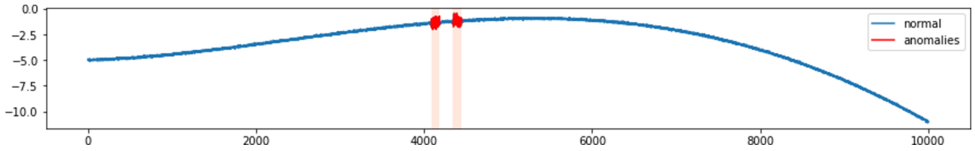




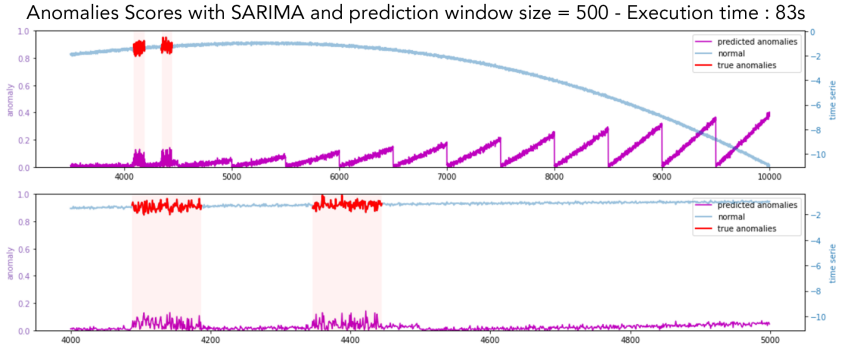


Appendix 9 : Influence of the prediction window for SARIMA.

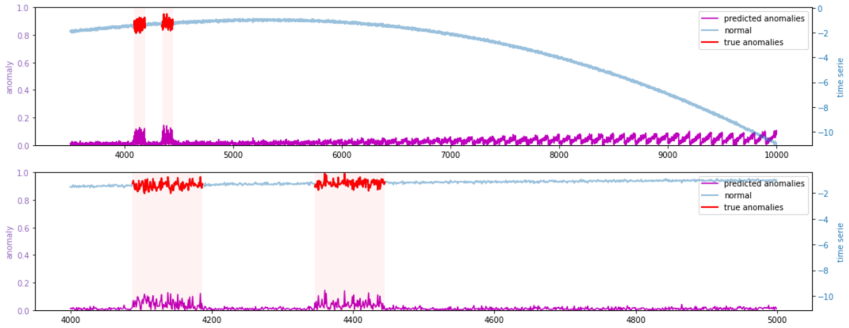
GutenTag dataset : poly-same-count-2



Anomalies Scores with SARIMA and prediction window size = 500 - Execution time : 83s



Anomalies Scores with SARIMA and prediction window size = 100 - Execution time : 180s



Anomalies Scores with SARIMA and prediction window size = 50 - Execution time : 260s