Platform for Anomaly Detection in Time-Series

Gheorghe Sebestyen, Anca Hangan, György Kovács, Zoltán Czakó

Technical University of Cluj-Napoca
Memorandumului 28, 400114, Cluj-Napoca, Romania
{gheorghe.sebestyen, anca.hangan, gyorgy.kovacs, zoltan.czako}@cs.utcluj.ro

Abstract: This paper presents a platform that integrates a number of functionalities necessary in the process of anomaly detection, from preprocessing towards various anomaly detection techniques and visualization methods. The purpose of this tool is to allow a developer to test, select and fine tune different algorithms that best fit anomaly detection in a given domain. To demonstrate the utility of the platform, we present a series of experiments done with different methods for anomaly detection on time-series and evaluate their results.

Keywords: anomaly detection, time-series, tools, data processing

1 Introduction

Monitoring of complex cyber-physical systems generates large amounts of data in the form of time series. Unfortunately, collected data may contain erroneous values, caused by various events (e.g. device defects, communication errors, environmental changes, etc.), which may affect the stability and controllability of the system. Therefore, a variety of methods were developed in order to identify and eliminate this so called "anomalies" [1] [2].

Anomaly detection in time series uses various methods [3] such as basic ones inspired from signal processing, methods that are based on time, space or functional correlations, methods based on system theory, or on pattern recognition. These methods perform differently on different data sets and for different types of anomalies. There is no "universal recipe" that works in all cases. Data analysts must choose the right approach for anomaly detection, taking into consideration the characteristics of data, the characteristics of the monitored system and the type of anomaly.

In this paper we propose a platform containing a set of tools that can aid in deciding which anomaly detection algorithm works best for a particular problem, by allowing the data analyst to try different anomaly detection approaches and then compare their results. Furthermore, we present a set of experiments in which we focus on anomaly detection techniques for time series data that contain different types of anomalies.

For the rest of this paper, we consider that analyzed data has the following form: $X = \{x_t \in \mathbb{R} : \forall t \geq 0\}$. Moreover, we will use the following functional definition

of an anomaly: "An anomaly is a data point or set of data, which is significantly different from all other data points or sets." This means that in order to define an anomaly, one must first have a notion of nominal (or normal) data. The term anomaly is relative and can not be applied to a data-point independently of any dataset.

The following chapters are structured as follows: In Section 2 we describe related work on the subject. Next, in Section 3 we give a brief description of the developed platform and its features. In Section 4 we describe the types of anomaly detection methods provided by our platform. We will test these methods in Section 5. This will be followed by conclusions and discussion on the future research directions.

2 Related Work

Most methods for anomaly detection are developed for a given field or have a specific application. In [4] and [5] the authors developed methods to detect anomalies in airplane data. In [6] similar methods were developed for IoT. In [7] methods for detecting anomalies in Big Data are presented. In [8] methods are used to detect anomalies in industrial machinery.

In [3] a group of algorithms and techniques are described and categorized. The authors found that while some algorithms in different fields are very similar, most algorithms are hard to generalize. They also note, that numerous formulations of anomaly detection problems are not sufficiently explored, i.e. it is not known how well some algorithms perform in a field that was not intended for that algorithm.

Efforts to bundle up different anomaly detection algorithms have already begun. In [9] the authors introduced an open source, generic framework for detecting anomalies in large scale time-series data.

In [10] a platform is proposed, that offers tools for data visualization, filtering and classification for a variety of data formats, including but not limited to time series data.

We believe that there is a real advantage in a platform that offers a variety of anomaly detection methods to the user. One can test the performance of a number of anomaly detection algorithms for some given test data. By having as many implementations of these algorithms as possible, the user can easily test as many algorithms as she wants with minimal effort and cost.

3 The Platform

Our platform includes a set of tools for data acquisition, preprocessing, anomaly detection and plotting. We envision the anomaly detection process as a workflow with configurable and interchangeable nodes in which the user has access to intermediate results. Figure 1 shows the envisioned workflow. The workflow has several types of nodes that the user includes in the custom anomaly detection process as follows:

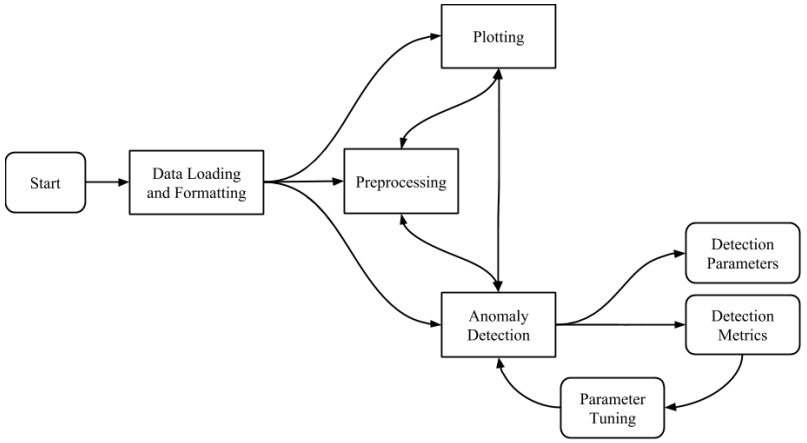

Figure 1
Workflow of the classification process.

- Data Loading: The user starts by loading in some data that she wishes to analyze. Because data can be stored in many different formats, the platform was designed to work with excel, Comma Separated Values (CSV) and Attribute-Relation File Format (ARFF) files, alongside MAT files.

- Plotting: The user can choose to visualize the data using the visualization tools provided by the platform. We provide tools both for time-series and non time-series data. For time-series data we provide tools for plotting, histograms and Fourier decomposition.

- Preprocessing: The user can apply transformations to the dataset either to smooth out the data or to emphasize some of its characteristics in an attempt to improve the classification results. For this we provide high-pass, low-pass and band-pass filters.

- Anomaly Detection: Next, a user can try out an anomaly detection algorithm from a collection of algorithms. Each one will have a set of parameters. These can be fine tuned to improve the classification quality. The quality of the classification can be measured using classification metrics. The confusion matrix can be generated as well as some derived metrics such as precision and recall.

When the user is satisfied with the classification, she can use the parameters of the classification method either for future tests or to implement a specialized system for monitoring anomalies.

4 Anomaly Detection Techniques

In this section we give short descriptions of some of the possible anomaly types. For each anomaly type we present some methods for detecting it.

In principle, anomaly detection may be derived from more classical methods such as system identification or classification. In the first case the methods try to identify a "nominal" behavior model for a system (or dataset) and detect anomalies as points or value sequences that don't fit with the model. In the second case a given classification method (from artificial intelligence field) is applied in order to label points as nominal or anomalous. Anomaly detection is slightly different from these at least in one regard: anomalies are generally rare, and comprise a very small percent of the data. Therefore the process of training the classifiers is sometimes difficult or even impossible.

Some datasets do not even contain anomalies, and algorithms are expected to learn the nominal functioning of the system from "clean data". We call a dataset that doesn't contain anomalies a clean dataset.

Depending on the moment when the detection is performed we can distinguish between on-line and off-line methods. Offline methods work on prerecorded datasets; usually they perform better, since for each data point we can use both past and future values and the processing time is not limited by the sampling period. The on-line methods are used for real-time detection, when only the past values are available and the detection algorithm must be performed in the interval between to samples. Sometimes the detection is delayed a few sampling periods in order to have values before and after the considered sample.

Next, a number of anomaly types are presented and also some techniques used for their detection.

4.1 Outliers

Given a dataset an outlier is a singular value (sample) that does not fit with its neighbours, because the value breaks somehow a given correlation with the neighbouring values. For this purpose, we define a function $label$ that labels a data-point $x \in X$ either as anomalous or nominal:

$$c_x = label(x, X)$$

$$c_x \in \{Nominal, Anomaly\}$$

4.1.1 Bounded Method

A very intuitive method would be to label elements as anomalies by setting global lower and upper bounds. This can be used to detect obvious anomalies such as extreme temperature levels or very high blood pressure.

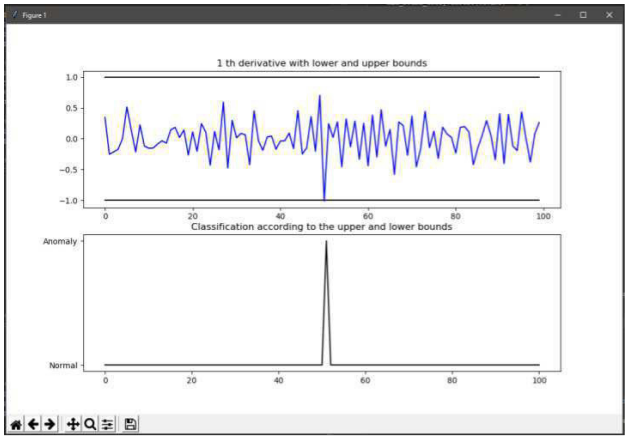

Figure 2
A signal with a higher than average disturbance in the middle is detected by the method. Forward difference is used as an approximation for the derivative.

In the platform we provide a method that classifies a point as an anomaly if it is outside certain bounds. This can be done even with any derivative of the function. We call this *Bounded Derivative Method*. BDM is defined as:

$$BDM(x) = \begin{cases} \text{Nominal} & \text{if } B^+ > f(x)^{(n)} > B^- \\ \text{Anomaly} & \text{otherwise} \end{cases}$$

where B^+ and B^- are the upper and lower bounds, and $f(x)^{(n)}$ is the n-th derivative of the signal. An example of the usage can be seen in Figure 2. The interval of "normality" may be set by the designer or it can be automatically detected using a dataset without outliers.

A more adaptive method, which can be applied for time-series with very slow or seasonal variations could set local bounds available for a given period or seasonal intervals of time. The same technique may be used for system that have multiple quasi-stationary stages.

Even though this detection is trivial, many existing real systems rely on this technique for detection.

4.1.2 Model Distance Methods

If the previous methods are not applicable, because of the complex nature of the time-series, more elaborate approaches can be used.

Since in many cases a time-series is generated by a generative process (e.g. the output of a linear system), one could accurately describe the underlying process, and create a model of the system:

$$x_t^* = f(t)$$

where x_t^* is the predicted value of x at time t.

Given such a model, the detection algorithm can label an anomaly based on the distance between an actual value and its prediction. Because the noise may influence the actual value a threshold must be established. If the distance (in absolute value) between the predicted value x_t^* and the actual value x_t is greater than some threshold d_{max}, x_t than it can be considered an anomaly:

$$p(x_t) = \begin{cases} \text{Anomaly} & \text{if } |x_t - f(t)| \geq d_{max} \\ \text{Nominal} & \text{otherwise} \end{cases}$$

The predictor of a time-series can be obtained using (auto)regression techniques or system identification methods (if the input of the system is also available). Although such a model is highly useful, sometimes in practice, in order to make a prediction, it is good enough to consider only the neighboring points.

Next we present some model distance methods implemented in our platform:

Median Distance Method This method is known as the *Double Sided Median Method* for anomaly detection [5]. By using a sliding window, we calculate the mean of the values inside the window, and if a given value is outside the allowed bounds, it is considered an anomaly.

$$DSMM(x_t) = \begin{cases} \text{Anomaly} & \text{if } |f(x_t) - mean(f(x_{t-k}), \ldots, f(x_{t+k}))| > d \\ \text{Nominal} & \text{otherwise} \end{cases}$$

An example can be seen in Figure 3.

Linear Approximation Method The linear approximation method is based on the assumption that the next point will fall on or near the line made up of the previous two points. The distance function is defined as the distance from a point to the line made up from the previous two points.

Auto-regressive Method Other regressive models, that use the past values to predict the new values are also used: $x_t^* = f(x_{t-1}, x_{t-2}, \ldots, x_{t-n}), n \geq 1$. If given that the data starts at $t = 0$ and $t - n > 0$, we use a sliding window approach, where we generate a prediction for the next value based on the actual old values. This is useful if we can model the time series using an auto-regressive model.

4.2 Change Point Detection

Change Point detection focuses on the underlying model of the process. The data is generated by a generative process $f(t, p)$, where $p \in \mathbb{R}^n$ are the parameters

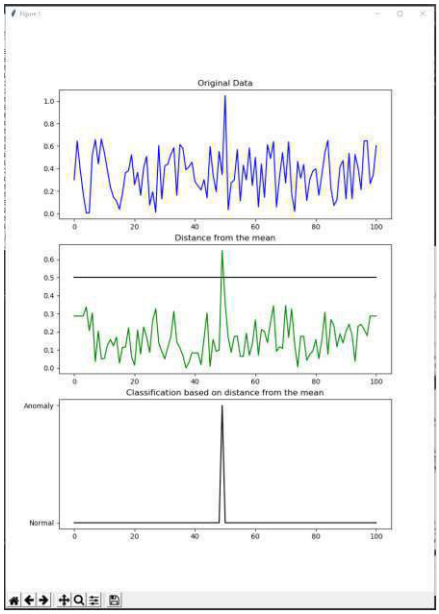

Figure 3
In the top figure, we can see the raw signal. In green we see the distance of each point from the mean of the sliding window. The black line from the middle plot is the distance limit. If the values falls outside the maximum distance, that point is considered to be an anomaly, as can be seen in the bottom plot.

of the model. That process is considered to be the nominal generative process. There is also an error $e(t)$ associated with the process. The error function is usually considered to be white noise.

$$x_t = f(t, p) - e(t)$$

This detection methods focuses on more long term changes compared to regular outlier detection methods. When a change in the system behavior is observed, it is considered an anomalous behavior. In other words, we constantly update the parameters of the process p_n, and compare it with the previous parameters p_{n-1}. p_n is considered an anomalous behavior if $|p_n - p_{n-1}| > d_p$, where $d_p \in \mathbb{R}$ is a threshold value.

An example of this is illustrated in Figure 4.

5 Experiments

In this section we will compare the results obtained with a number of anomaly detection methods.

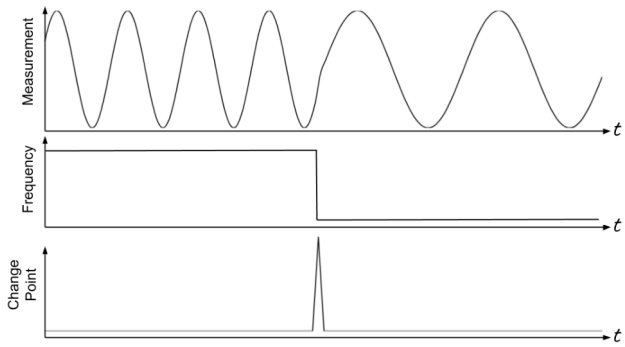

Figure 4
In the top most graph, we can see the observed process. In this case it is a pure sine wave. In the middle graphs we can see the coefficient of the model, which in this case is just a function of it's frequency, since it is enough to perfectly describe the process. We consider as anomaly either the change point, either all the points where the model is outside some bounds.

The figures presented throughout this section display the values from different datasets labeled as normal (blue dots) and outliers (red dots). The points were labelled by the authors of this paper. This may seem arbitrary, but we contend that it is a valid method for establishing ground truth because of the fact that the very definition of anomaly is vague and is difficult to express in mathematical terms, yet it can be very intuitive. This is similar to the difficulty of expressing what a cat looks like in an image, however one would have no difficulty whatsoever determining if a picture contains a cat by just looking at it. It is also important to mention that the labelling phase was completed before the experiments and the labels were not changed in any way afterwards.

In order to judge the validity of the methods, a number of metrics were used. For each method, we included the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) labels, as well as the calculated accuracy, precision and recall.

The accuracy represents the overall quality of the classification; it is just the percentage of labels that are correct. However since by definition most anomalies are rare events, they do not influence the final score as much. This is why this metric is not a good measure for the classification.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision refers to the number of relevant (anomalous) labels from the correctly identified selection.

$$precision = \frac{TP}{TP + TN}$$

Recall refers to the number of correct relevant (anomalous) points selected.

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

5.1 First Dataset - Concurrent Users

We will use the a training dataset from [9]. The dataset is a univariate time-series displayed in Figure 5. This time series is a periodic signal with a couple of extreme points that are considered anomalies. Most points are either too low, or too high. The results obtained are aggregated in Table 1.

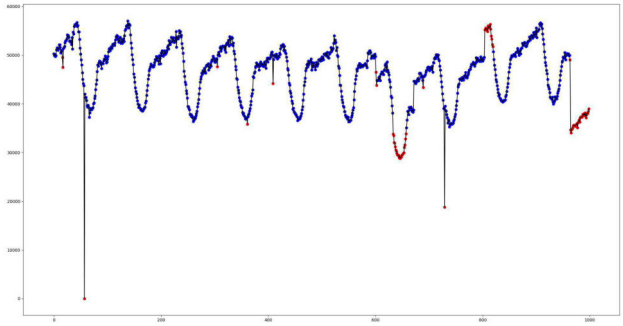

Figure 5
The x axis represents the timestamp of the measurement, and the y axis represents the number of concurrent users.

Table 1
Results of applying different methods to the first dataset.

Method	TP	TN	FP	FN	Accuracy	Precision	Recall
Bounded Derivative (d = 0)	32	912	0	55	0.9449	0.0339	1.0000
Bounded Derivative (d = 1)	6	908	4	81	0.9149	0.0066	0.6000
Median Method	8	896	16	79	0.9049	0.0088	0.3333
Linear Approximation	3	903	16	79	0.9069	0.0033	0.2500
First Order AR	1	904	8	86	0.9059	0.0011	0.1111

5.2 Second Dataset - Concurrent Users

The next dataset is also from [9]. For the second dataset we used a second univariate-time series similar to the previous one. This new one has clear change-point anoma-

lies, as well as the extreme values from the previous dataset. The shape of the graph and the labelled points are visible in Figure 6.

One could consider the first third of the data as anomalous behavior right until the change point. That is a valid interpretation of the data. However we decided not to go on that route, instead considering only the change point as an anomaly. The results are given in Table 2.

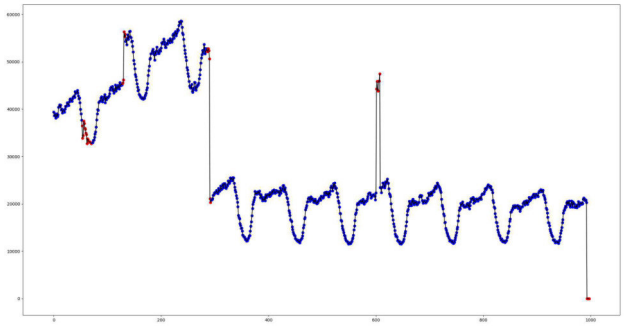

Figure 6
The x axis represents the timestamp of the measurement, and the y axis represents the number of concurrent users.

Table 2
Results of applying different methods to the second dataset.

Method	TP	TN	FP	FN	Accuracy	Precision	Recall
Bounded Derivative (d = 0)	42	692	261	3	0.7355	0.0572	0.1386
Bounded Derivative (d = 1)	7	952	1	38	0.9609	0.0073	0.8750
Median Method	20	944	9	25	0.9659	0.0207	0.6897
Linear Approximation	9	951	2	36	0.9619	0.0094	0.8182
First Order AR	1	947	6	44	0.9499	0.0011	0.1429

5.3 Third Dataset - Synthetic Data

The third dataset is a synthetic one and was generated by adding a white noise on top of a sine wave. Two points have been added that are outside the “normal” range of the signal. These two are considered anomalies, while the others are considered nominal points. The graph with the labels can be seen in Figure 7, and the result of the different methodologies can be seen in Table 3.

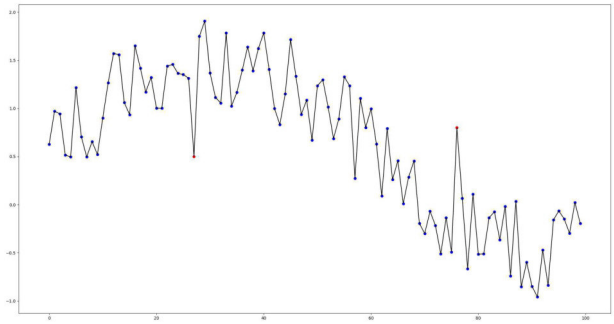

Figure 7
The axes have no real significance since this is just synthetic data that is obtained by adding together a signal with white noise and one with a sine wave.

Table 3
Results of applying different methods to the third dataset.

Method	TP	TN	FP	FN	Accuracy	Precision	Recall
Bounded Derivative (d = 0)	0	98	0	2	0.9800	0.0000	——
Bounded Derivative (d = 1)	1	97	1	1	0.9800	0.0102	0.5000
Median Method	0	96	2	2	0.9600	0.0000	0.0000
Linear Approximation	0	97	1	2	0.9700	0.0000	0.0000
First Order AR	0	96	2	2	0.9600	0.0000	0.0000

5.4 Fourth Dataset - CO2 emission

The last dataset is from [11]. It represents the CO2 (ppm) emitted by the Mauna Loa volcano between the years 1965 and 1980. The ground truth classification can be seen in Figure 8. We observed that most neighbouring points have a certain distance between them, while a small number of neighboring points are more closely bunched. We considered these points as anomalous. The classification metrics are aggregated in Table 4.

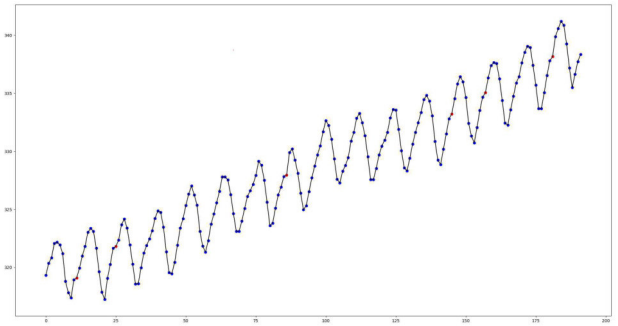

Figure 8
The x axis represents the months between 1965 and 1980, while the y axis represents the emitted CO2.

Table 4
Results of applying different methods to the fourth dataset.

Method	TP	TN	FP	FN	Accuracy	Precision	Recall
Bounded Derivative (d = 0)	0	186	0	6	0.9688	0.0000	——
Bounded Derivative (d = 1)	0	182	4	6	0.9479	0.0000	0.0000
Median Method	0	182	4	6	0.9479	0.0000	0.0000
Linear Approximation	0	181	5	6	0.9427	0.0000	0.0000
First Order AR	0	183	3	6	0.9531	0.0000	0.0000

Conclusions

This paper presents an anomaly detection platform that incorporates a number of methods needed in the process of anomaly detection. The platform is useful for finding the best method that fits for a given dataset. This tool will allow a designer to decide which anomaly detection technique should be integrated in a given real-world application.

The experiments showed that in some cases less complex methods (with less execution time requirements) can generate results that are similar to more complex and more time consuming methods. The platform was designed in a modular manner in order to allow easy integration of new methods. The detection process is built up as a sequence of modules which may be combined by the user in order to get the best results.

In the future we intend to add more complex detection techniques which use artificial intelligence methods for pattern recognition.

Another point of interest are the classification metrics themselves. While the traditional confusion matrix is useful, we noticed that most errors were “off by one” classifications. Many times we have seen algorithms detect an anomaly, but placed

it with one time-stamp before or after the ground truth. Since the detection of the anomaly may be more important than the exact pinpointing of it, some metrics could be devised to take this preference into account.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey", *ACM Computing Surveys 41,3*, 2009.

[2] M. Rassam, A. Zainal, and M. Maarof, "Advancements of data anomaly detection research in wireless sensor networks: A survey and open issues", *Sensors*, 2013.

[3] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temportal data: A survey", *IEEE Transactions on Knowledge and Data Engineering*, 2014. DOI: 10.1109/TKDE.2013.184.

[4] G. Silvestri, F. B. Verona, M. Innocenti, and M. Napolitano, "Fault detection using neural networks", in *IEEE World Congress on Computational Intelligence*, 1994. DOI: 10.1109/ICNN.1994.374815.

[5] S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: An application to sensor data", *Knowledge and Information Systems*, 2006. DOI: 10.1007/s10115-006-0026-6.

[6] T. J. Lee, J. Gottschlich, N. Tatbul, E. Metcalf, and S. Zdonik, "Greenhouse: A zero-positive machine learning system for time-series anomaly detection", *CoRR*, vol. abs/1801.03168, 2018.

[7] C. K. Maurya, D. Toshniwal, and V. Agarwal, "Anomaly detection via distributed sparse class-imbalance learning", in *International Conference on Machine Learning*, 2016.

[8] D. Dasgupta and S. Forrest, "Novelty detection in time series data using ideas from immunology", in *International Conference on Intelligent Systems*, 1996.

[9] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection", in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1939–1947.

[10] G. Sebestyen, A. Hangan, Z. Czako, and G. Kovács, "A taxonomy and platform for anomaly detection", *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pp. 1–6, 2018. DOI: 10.1109/AQTR.2018.8402710.

[11] R. Hyndman, *Time series data library*, https://datamarket.com/data/list/?q=provider:tsdl, Accessed: 2018-11-12.