Acta Univ. Sapientiae, Informatica **11**, 2 (2019) 113–130

DOI: 10.2478/ausi-2019-0008

Evaluation metrics for anomaly detection algorithms in time-series

György KOVÁCS
Technical University of Cluj-Napoca, Romania
email: Gyorgy.Kovacs@cs.utcluj.ro

Gheorghe SEBESTYEN
Technical University of Cluj-Napoca, Romania
email: Gheorghe.Sebestyen@cs.utcluj.ro

Anca HANGAN
Technical University of Cluj-Napoca, Romania
email: Anca.Hangan@cs.utcluj.ro

Abstract. Time-series are ordered sequences of discrete-time data. Due to their temporal dimension, anomaly detection techniques used in time-series have to take into consideration time correlations and other time-related particularities. Generally, in order to evaluate the quality of an anomaly detection technique, the confusion matrix and its derived metrics such as precision and recall are used. These metrics, however, do not take this temporal dimension into consideration. In this paper, we propose three metrics that can be used to evaluate the quality of a classification, while accounting for the temporal dimension found in time-series data.

1 Introduction

Anomaly detection is the process of identifying erroneous data in big data sets, in order to improve the quality of further data processing. An anomaly detection method classifies data into normal and abnormal values. The selection

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 68R15
Key words and phrases: anomaly detection, classification, evaluation metrics

of the best detection method greatly depends on the data set characteristics. Therefore, we need metrics to evaluate the performance of different methods, on a given data set.

Traditionally, in order to evaluate the quality of a classification, the confusion matrix, or one of its derived metrics is used. These metrics work well when the data set does not have a temporal dimension.

The anomaly detection task has certain particularities when it comes to time-series data. The temporal dimension that may be lacking in other types of data sets can be taken into account in order to improve the evaluation of these methods.

In this paper we propose some evaluation metrics which are more appropriate for time series. The basic idea of the new metrics take into consideration the temporal distance between the true and predicted anomaly points. This way, a small time shift between the true and the detected anomaly is considered a good result as opposed to the traditional metric that will consider it an erroneous detection.

Through a number of experiments, we demonstrate that our proposed metrics are closer to the intuition of a human expert.

The remainder of this paper is organized as follows: Section 2 discusses how time-series classification is used in the field and what metrics are used to evaluate the quality of classifications. Section 3 discusses the notation that we will be using in this paper going forward. The anomaly detection problem is defined for time-series data. We also define some prior requirements that we expect to be true for a time-series data metric that takes temporal distances into account. Section 4 presents the classification metrics we propose which are evaluated in Section 5. We check if the assumptions from Section 2 hold for our metrics. We also compare them with the traditional confusion matrix derived metrics such as accuracy, precision, and recall. We also present the result of applying our methods to real-world data. Section 6 concludes the paper.

2 Related work

Anomaly detection in time-series data is an important subset of generic anomaly detection. Much work has been done in developing anomaly detection methods. Some work has also been done in developing better metrics.

In many applications, it is more efficient to do feature extraction on the time-series data, and do classifications based on those features rather than on the actual time-series, as is the case for [7]. This is due to the fact the in many applications the volume of time series data is large and multi-dimensional. It is not easily analyzed and in many applications where speed is important, it is not practical to run algorithms directly on the raw data. Instead, the time series is split into segments, and for each segment features such as mean value, maximum and minimum amplitude and so on are calculated. Here classical clustering methods such as K-Nearest Neighbor can be used to classify each segment of time-series data.

The confusion matrix is generally used in the context of times series classification, which is the case in [1]. In [3] the authors use the confusion matrix explicitly as an input to train the classification model.

Better metrics for time-series have been proposed. In [2] the authors propose a metric that can differentiate between the generative processes of the time-series data. In [4] the authors propose a number of metrics such as Average Segmentation Count (ASC), Absolute Segmentation Distance (ASD) and Average Direction Tendency (ADT). These metrics were developed for evaluating a segmentation of a time-series, but they can be used just as well for evaluating the quality of anomaly detection. We will slightly modify the names of ASC and ASD by replacing segmentation with detection. In the experiments section, we will use these metrics Average Detection Count (ADC) and Absolute Detection Distance (ADD) and compare them with our metrics.

3 Problem statement

3.1 Notation

In order to express concisely the ideas presented in this paper, we will define the main concepts of anomaly detection and use the notation presented in this chapter for the following chapters as well.

This paper discusses concepts related to time-series data. By time-series data we mean an ordered set of real values that are indexed by natural numbers. We will not be discussing continuous values, since in practice we measure by sampling.

$$X = \{x_0, x_1, x_2, \dots, x_n\}, x_t \in \mathbb{R}$$

The main focus of this paper are classifications. The set of class labels, which will be referred to as a *classification*, is similar to X, the difference being that while X consists of real values, C consists of binary values $\{0, 1\}$. We will consider values labelled as 0 as being normal values, and values labelled as 1 being anomalous values.

$$C = \{c_0, c_1, c_2, \ldots, c_n\}, c_t \in \{0, 1\}$$

The classification is generated by a classifier function $\mathcal{C}$. The classifier function takes an $x_t \in X$ value, with a range of size w around it, and generates a classification value c_t. Thus, this classifier can be used for a sliding window classification.

$$\mathcal{C} : X^{2w+1} \to C$$

$$c_t = \mathcal{C}(x_{t-w}, \ldots, x_t, \ldots, x_{t+w})$$

For a simplified example, consider the following time-series data:

$$X = \{8, 8, 8, 8, 42, 8, 8, 8, 8\}$$

We classify this data using the following classifier:

$$\mathcal{C}(x_t) = \begin{cases} 1 & \text{if } x_t > 10 \\ 0 & \text{otherwise} \end{cases}$$

In the example given $w = 0$. Thus the classifier only looks at one point for each classification. The result is the following classification:

$$C = \{0, 0, 0, 0, 1, 0, 0, 0, 0\}$$

We can represent the classification visually in Figure 1 as a line, where each point represents a classification. At the points where the classification value is 1, we draw a short vertical line through the horizontal line. We use this notation because our base assumption is that anomalous values are a disproportionately small subset of the whole set of values.

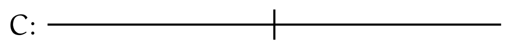


Figure 1: A classification C represented by a line with vertical lines where the value of C are 1.

Next, we define the classification evaluation problem. Supervised learning is a machine learning task with the purpose of reproducing a function by looking at example inputs and outputs. Given two or more potential candidate functions it is important for such an algorithm to be able to rank such functions.

In order to decide which one approximates the original function better, some metric is used.

This problem can be expressed as a comparison of classifications generated by different classifiers. We consider the target classifier $\mathcal{C}_0$. This is the function that we would like to reproduce. Given a number of different classifiers $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ we would like to find which one approximates $\mathcal{C}_0$ the most.

In order to do this, we compare the classifications generated by them given the same training data X. We will use the graphical representation from Figure 2.

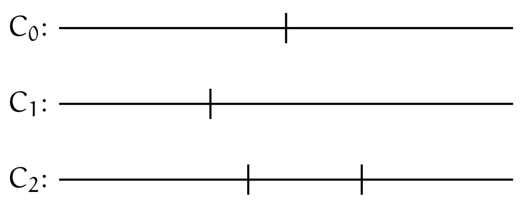


Figure 2: A comparison of three classifications, the first one being the target classification C_0 and the rest are regarded as the candidate classifications. One can see that C_1 identifies the anomaly prematurely while C_2 identifies two anomalies, one prematurely and one with a delay

We compare the classifications using a metric $m : \mathbb{C} \to \mathbb{R}$, $C_i \in \mathbb{C}$. The metric produces a score by comparing the given classification with the target classification. If a classification C_a is considered better than another one, say C_b, the metric would produce a better score for C_a. If a metric produces a better score for a classification, we consider that classification as better.

As an example, suppose we have the classifications from Figure 2. We will use a simple metric, that counts the number of anomalous points in each classification and computes the difference between that classification and the target classification.

We define the count function as the sum of all the elements of a classification. This effectively counts the number of anomalies because they are represented by the number 1, while the normal values are represented by the number 0.

$$\text{count}(C_i) = \sum_{j=1}^{n} c_j, c_j \in C_i$$

Next, we simply calculate the difference between the number of anomalies in the target classification and the candidate classification.

$$m_{\downarrow}(C_i) = |\text{count}(C_0) - \text{count}(C_i)|$$

Using this classification, we can see that the score for C_1 is $m_{\downarrow}(C_1) = 0$ and the score for C_2 is $m_{\downarrow}(C_2) = 1$. We can say that the first classification is better than the second one, since it has a lower value. This is represented by the subscript arrow that is pointing down. A metric where a higher value is better is denoted by a little arrow pointing up.

The examples presented in this chapter are simplistic and are only used to familiarize the reader with the notation that will be used for the remainder of this paper.

3.2 Prior requirements

We give examples with a target classification and a number of candidate classifications. We rank the classifications using our intuition. We would like a classification metric that can capture that intuition. These assumptions may or may not hold for certain applications.

Each of these situations will be tested both by the existing metrics used, and also our proposed metrics. In Section 5 we aggregate all the score data and show if the given metric does indeed respect these requirements.

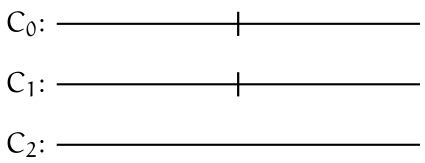
Detection The first requirement is to rank a classification that finds an anomaly higher than one that doesn't. The graphical representation can be seen in Figure 3a. Because C_1 correctly detects the anomaly, whereas C_2 does not.

False Detection The second requirement is that if there is indeed no anomaly, we would consider the classification that doesn't detect an anomaly as the better one. The graphical representation can be seen in Figure 3b. Because the target classification does not contain anomalies and C_2 falsely detects an anomaly, we can say that it is the worst from the two.

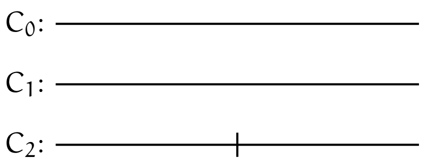
Less Wrong Whenever we have two classifications that correctly predict the anomaly, an ideal metric would choose the one with the fewer incorrect classifications, as presented in Figure 3c. Because both C_1 and C_2 correctly detect the anomaly, C_1 is considered better because it has less errors than C_2.

Near Detection Here we are starting to enter controversial territory. Given that we use time-series, we will hold the points near the point of interest in higher regard than those farther away. We consider that a classification that almost detects the anomaly correctly, is better than one that doesn't detect the anomaly at all. The graphical representation can be seen in Figure 3d. While none of the classifications manage to exactly detect the anomaly in the right place, we consider C_1 as better because at least it did detect something relatively close to the anomaly, while C_2 did not at all.

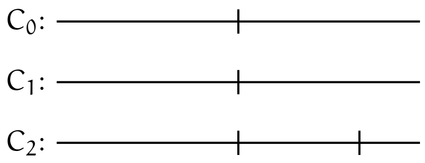
Closeness Going further, we consider that the closer the detection is to the actual anomaly in the target classification, the better that classification is. The graphical representation can be seen in Figure 3e. While both classification missed the anomaly, C_1 detected an anomaly closer to the target one than C_2.



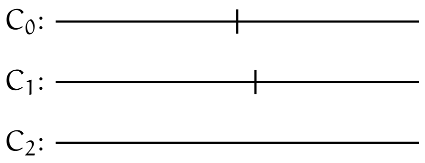
(a) Detection



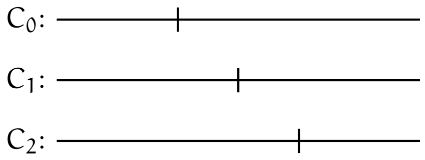
(b) False Negative



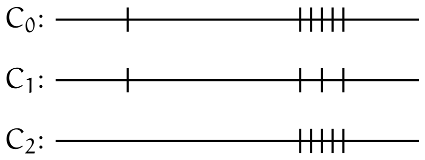
(c) Less Wrong



(d) Near Detection



(e) Closeness



(f) Locally Perfect vs Globally Good

Figure 3: Visualisation of the requirements. In each case the top classification (C_0) is the target classification and in these examples middle classification (C_1) is considered to be a better classification than the lower one (C_2).

Locally Perfect vs Globally Good Lastly, we introduce the principle of being Globally Good instead of Locally Perfect. This rule emphasizes the fact that we can have clusters of anomalies. Each cluster can have only one anomaly or multiple, but in close proximity to each other. This rule assumes that it is better to discover each cluster, rather than perfectly match every single anomaly from one single cluster. In the example from 3f, we can see that C_0 has two clusters, one with one single anomaly, and one with five close anomalies. We consider C_2 worse than C_1 even though it perfectly described the anomalies from the second cluster, because it failed to detect the first cluster.

4 Proposed metrics

4.1 Temporal distance method

This method consists of calculating the sum of all distances between anomalies from the two classifications. This method is similar to the ADD metric from [4]. The difference being that while in ADD we look only in the proximity of the detection, while our method looks at the closest detection, regardless of proximity. To this end we define a function that calculates the distance between each anomaly of the first classification and the corresponding closest anomaly from the second classification, $f_{closest} : \mathbb{C}^2 \to \mathbb{R}$.

Next we can define our method by using the function described above in the following manner:

$$\mathrm{TD}_{\downarrow}(C_i) = \mathrm{TTC} + \mathrm{CTT}$$

where TTC stands for Target To Candidate and is given by $f_{closest}(C_0, C_i)$, and CTT stands for Candidate To Target and is given by $f_{closest}(C_i, C_0)$. C_0 stands for the target classification and C_i stands for the candidate classification.

Note that lower values produced by this method are better than higher ones. This is represented by a little downward pointing arrow.

We calculate both the sum of all the distances of the closest anomalies from the candidate classification to the target classification (CTT), and the sum of all distances of the closest anomalies from the target classification to the candidate one (TTC). By adding these two together, we have a metric that punishes false negative values and false positive values. TTC punishes false negatives and CTT punishes false positives.

A graphical visualization of the metric can be seen in Figure 4. We can see that C_0 has two anomalies, but C_1 only has one. Thus the closest anomaly to

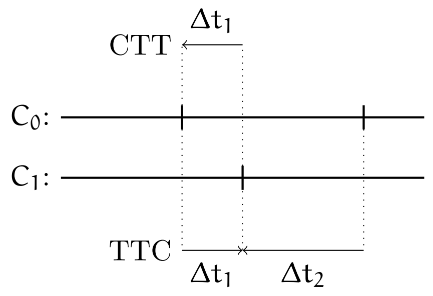


Figure 4: Visualization of the Temporal Distance metric. CTT $= \Delta t_1$ and TTC $= \Delta t_1 + \Delta t_2$

both anomalies from C_0 is the single anomaly in C_1. Because the two temporal distances are Δt_1 and Δt_2 respectively, TTC $= \Delta t_1 + \Delta t_2$. However, from the perspective of C_1, the closest anomaly to its only anomaly is the first from C_0. Only that one is taken into account, thus CTT $= \Delta t_1$. Finally the resulting value calculated by the metric is $\text{TD}_{\downarrow}(C_1) = 2\Delta t_1 + \Delta t_2$. One can see that the best possible value for this metric is 0.

We also define a variation of this metric that we dubbed the Squared Temporal Distance. STD is defined similarly to TD, except when adding up the distances they are first squared. This is done in order to punish larger distances more than smaller ones. For example the value of STD for the given example is $2\Delta t_1^2 + \Delta t_2^2$.

4.2 Counting method

The next method is also similar to the ADC method defined in [4]. However our method is more analogous to a forgiving decision matrix, that counts close detections as true positives.

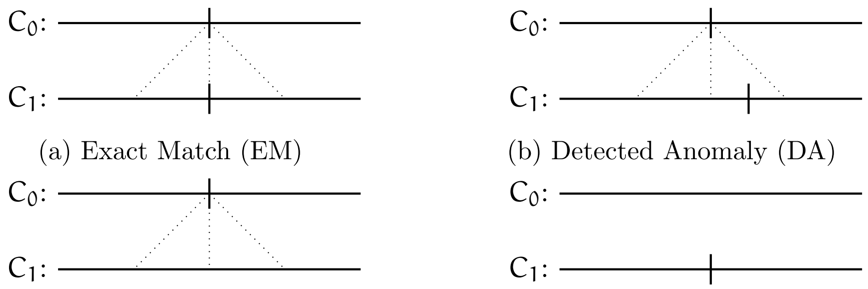
The counting method counts the occurrence of four situations. The situations of interest are pictured in Figure 5, and are as follows:

1. **Exact Match (EM)** Figure 5a. If the anomaly is from the candidate classification matches exactly the anomaly in the target classification, we call that an exact match.

2. **Detected Anomaly (DA)** Figure 5b. If the candidate anomaly does not match exactly the target anomaly, a range is considered. If the anomaly is withing that range, it is considered as a detection. However, that candidate anomaly will also be counted up as a false anomaly.

3. **Missed Anomaly (MA)** Figure 5c. If there is no anomaly present in the expected range of the target anomaly, we count it as a missed anomaly.

4. **False Anomaly (FA)** Figure 5d. Every normal target value that has an associated anomalous candidate value is counted as a false anomaly.



(a) Exact Match (EM) (b) Detected Anomaly (DA)

(c) Missed Anomaly (MA) (d) False Anomaly (FA)

Figure 5: Relevant situations that are each counted.

These four values can be used further to define derived metrics. We define two such metrics here:

Total Detected In Range We calculate the ratio of correctly detected anomalies to the number of total anomalies. With the maximum score of 1 and minimum of 0, this metric is similar to recall as derived from the confusion matrix. We can also call this metric "forgiving recall".

$$\mathrm{TDIR}_{\uparrow} = \frac{\mathrm{EM} + \mathrm{DA}}{\mathrm{EM} + \mathrm{DA} + \mathrm{MA}}$$

Detection Accuracy In Range We calculate the ratio of correctly detected anomalies to the total number of detected anomalies. With the maximum score of 1 and minimum of 0, this metric is similar to precision as derived from the confusion matrix. We can also call this metric "forgiving precision".

$$\mathrm{DAIR}_{\uparrow} = \frac{\mathrm{EM} + \mathrm{DA}}{\mathrm{EM} + \mathrm{DA} + \mathrm{FA}}$$

We could have implemented "forgiving" versions of the metrics defined for the confusion matrix, and we could have defined some new ones, like the ratio of exact matches to detected anomalies. However in the interest of conciseness, we will only consider the two defined above.

4.3 Weighted method

The weighted method can be seen as a combination of the previous two methods. For each anomaly of the target classification we calculate a weight that is given by a function that takes the distance of the candidate anomaly to the target anomaly and produces the weight. In Figure 6 we calculate a weight w for an anomaly. In the example a bell curve function is used. Any function based on distance can be used as long as it is monotonically decreasing.

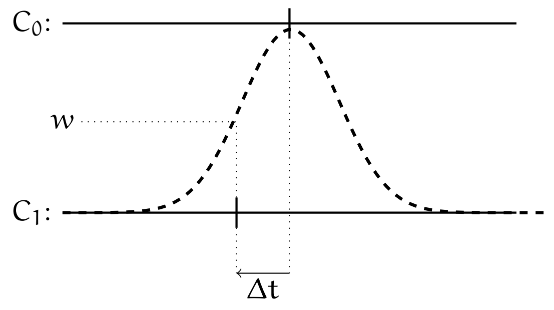


Figure 6: Weighted metric with a gaussian function, $w = f(\Delta t)$, where f describes a bell curve.

We denote with WS the sum up all the weighted values produced for each target anomaly. Note that we only take into consideration the closest candidate anomaly. We also count up all the false anomaly cases FA, similar to the previous section.

We define the **Weighted Detection Difference Metric** using the WS and FA. We just scale the FA by some factor and subtract it from the WS.

$$\text{WDD}_{\uparrow} = \text{WS} - w_f * \text{FA}$$

where w_f is the weight of the false anomalies. Other functions were also considered such as a linear function:

$$f(\Delta t) = 1 - \frac{\Delta t}{t_{max}}$$

or a variant that punishes outliers equally:

$$f(\Delta t) = \begin{cases} 1 - \frac{\Delta t}{t_{max}} & \text{if } \Delta t < t_{max} \\ -1 & \text{otherwise} \end{cases}$$

5 Analysis and experiments

5.1 Requirements evaluation

We took each of the six rules defined in Section 3.2, and we turned it into synthetic data. We did this by considering 101 points for each Classification. For each point we assigned a one where the figure had a vertical line, and assigned a zero for the rest of the points. In fact the figures that appear in Section 3.2 where generated from these synthetic datasets. We used the synthetic datasets and calculated the scores produced for all the methods described in the previous chapter alongside classical methods. The results are aggregated in Table 1.

Table 1 is the cross product of the metrics we used and the rules we defined in Section 3.2. We differentiate between four possible outcomes. Each outcome is represented by a special character:

✓ We represent cases where the metric strictly respects the rule set out by us via a checkmark. Effectively this means that the metric produced a better result for the first candidate classification than for the second one. The actual value produced by the classification can be larger or smaller, depending on the metric used.

= Equality does not respect the rule we set out in Chapter 2. However we decided to show it explicitly because it gives better insight into the workings of the metric. While we do not consider equality cases to be an instance of a successful quality evaluation, we consider them as cases where the metric can not tell the difference between two candidate classifications.

× In cases where the metric gives a strictly better score to a worse candidate classification, we consider that as a broken rule. A metric that would fit the rules we laid out would never break any rules.

- There are cases where a metric can not be calculated. Otherwise stated, there are classifications that yield scores that can not be expressed by real numbers. These cases arise because the metric can use the number of anomalies detected in order to divide some other number. If there are no anomalies, we can not perform that operation. We call this situation undecidable and use a dash to denote that situation.

In the second example we produced the ranking in a similar fashion to the previous example. The actual change point happens in the third group of anomalies from C_0. We consider that only the change point is an anomaly. The rest of the anomalies are outliers. Outliers can be found both before and after the change point.

The metrics that best matched the imposed ranking this time were Recall, ADD, STD and DAIR. In this particular example the classical methods had similar distances to the proposed metrics. We believe that this is because of the fact that in this particular example, all of the anomaly groups were made up of sequential anomalies.

Consider a classifier that is always a few time-samples behind with the classification. If all anomalies are point anomalies, all candidate anomalies would miss the target anomalies and a classical metric would produce a bad score. Now if the anomalies were not point anomalies, but were a continuous intervals, even though the candidate intervals of anomalies were shifted, most anomalies would still overlap, thus producing a better score.

$\mathfrak{m}$	Det	FDet	LWrong	NDet	Close	LP vs GG
Accuracy$_\uparrow$	✓	✓	✓	$\times$	=	$\times$
Precision$_\uparrow$	✓	-	=	=	=	$\times$
Recall$_\uparrow$	-	-	✓	-	=	=
ADC$_\downarrow$	✓	=	=	✓	✓	✓
ADD$_\downarrow$	-	-	=	-	✓	=
TD$_\downarrow$	-	-	✓	-	✓	✓
STD$_\downarrow$	-	-	✓	-	✓	✓
TDIR$_\uparrow$	✓	-	=	✓	✓	✓
DAIR$_\uparrow$	-	-	✓	-	✓	=
WDD$_\uparrow$	✓	✓	✓	✓	✓	✓

Table 1: For each rule defined in Section 2 we verify if the metric respects the relation.

The table shows that while some of the proposed metrics may sometimes be unable to distinguish between two classifications or evaluate an answer, they never give erroneous results. The same can not be said of Accuracy or Precision.

From our experiments, while WDD obtained the best results, the efficacy of the metric is very much dependent on the parameters used. By modifying the parameters we can get counter intuitive results.

5.2 Real data example

In order to further test the quality of our metric we will apply them to some example time-series traffic data. Two datasets will be considered. The first only has outlier points while the other also contains a change point.

Both datasets and classifiers are used in [8]. The classifiers are described in chapter 4 of that article. The classifications $\{C_1, \dots, C_5\}$ are generated by the classifiers {Bounded Derivative (d = 0), Bounded Derivative (d = 1), Median Method, Linear Approximation, First Order AR}. The classifiers will not be discussed in this paper.

The first dataset is from [5], and the second one from [6]. The classification diagrams of the aforementioned datasets can be seen in Figure 7 and Figure 8 respectively. We evaluate each classification with each of the metrics used in Table 1. We also add another metric that ranks each classification according to the subjective opinions of the authors. We would like that all metrics generate a similar order to the one imposed by us.

The imposed ranking is done by assigning a number starting from 1 to each classification, where 1 is considered the best classification, and 5 is considered the worst. Next we compare that ranking with the ranking generated by the metrics. We consider the classification with the best result as the one with ranking of 1, the second best with 2 and so on. This step can be checked manually in the table. Finally we calculate the distance between the imposed ranking and the ranking generated by the metric. The lower the distance, the better the metric performed. The distance is calculated by summing the differences between the two rankings.

$$\text{Distance} = \sum_{i=1}^{5} |r_i - \hat{r}_i|$$

where r_i is the ranking of the classification C_i generated by a metric m and $\hat{r}_i$ is the imposed ranking of the given classification C_i.

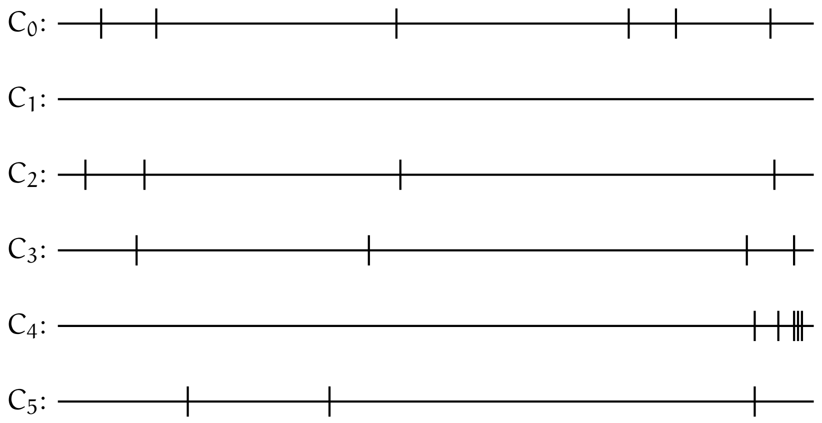


Figure 7: CO_2 emissions

m	$m(C_1)$	$m(C_2)$	$m(C_3)$	$m(C_4)$	$m(C_5)$	Distance
Accuracy$_\uparrow$	0.9688	0.9479	0.9479	0.9427	0.9531	9
Precision$_\uparrow$	0.0000	0.0000	0.0000	0.0000	0.0000	10
Recall$_\uparrow$	-	0.0000	0.0000	0.0000	0.0000	6
ADC$_\uparrow$	0	4	4	5	2	5
ADD$_\downarrow$	-	9	24	27	12	2
TD$_\downarrow$	-	8	99	490	132	0
STD$_\downarrow$	-	2048	1561	60538	2646	2
TDIR$_\uparrow$	0.0000	0.6667	0.6667	0.1667	0.3333	1
DAIR$_\uparrow$	-	0.5000	0.5000	0.1667	0.4000	1
WDD$_\uparrow$	0.0000	-3.5000	-3.9000	-43.3000	-6.1000	8
Ranking$\downarrow$	5	1	2	4	3	0

Table 2: CO_2 emissions. Note that all missing values are considered to be the worst scores.

For the first example we considered C_0. None of the classifiers managed to pin down the anomalies exactly. While all of them were off by some margin, some of them are clearly worse than others. For example, C_1 didn't detect any anomalies, while C_4 detected a cluster of them towards the end, where only one anomaly exists. Looking at the table of results, we can see that precision and accuracy can not tell the difference between these classifications. However, we would argue that C_2 is the clear winner. While the detection of the anomalies are off by one or two time samples, they are still in the neighborhood of the true anomalies. Only the fourth and fifth anomalies are missed by it. 2.

The metrics that best matched our ranking were TD, TDIR, DAIR. STD and ADD matched but they are also good classifications. This example shows the potential instability of the WDD method, that produced a ranking that is as bad as the ones produced by the confusion matrix.

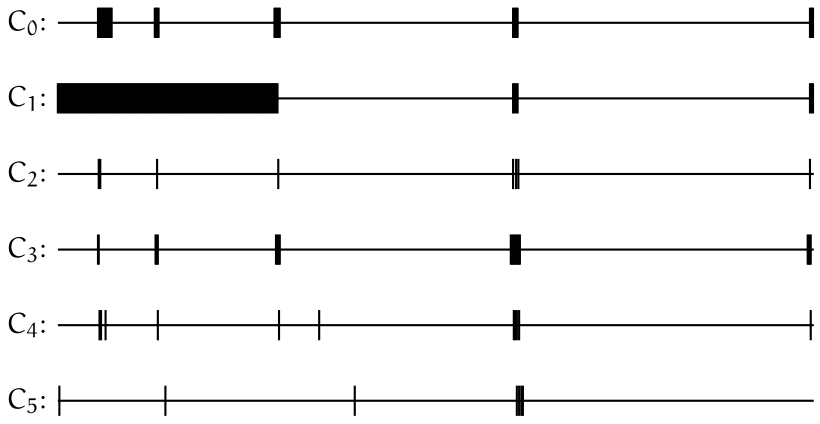


Figure 8: Concurrent users – Change point

6 Conclusion

In this paper we tackled with the problem of qualitative metrics applied to anomaly detection in time-series data. We concluded that classical metrics such as Accuracy, Precision and Recall do not take into consideration the time dimension of time-series data, in which near matches might be just as good as exact matches, or at least they are better than complete misses.

We defined the problem in more rigorous terms, and provided some requirements that we believe a good metric should meet. Next we defined some new metrics. We checked whether or not our proposed metrics respect the requirements set out by us previously. We also compared the performance of our

$\mathfrak{m}$	$\mathfrak{m}(C_1)$	$\mathfrak{m}(C_2)$	$\mathfrak{m}(C_3)$	$\mathfrak{m}(C_4)$	$\mathfrak{m}(C_5)$	Distance
Accuracy$_\uparrow$	0.7354	0.9609	0.9659	0.9619	0.9498	4
Precision$_\uparrow$	0.9333	0.1555	0.4444	0.2	0.0222	8
Recall$_\uparrow$	0.1386	0.875	0.6896	0.8181	0.1428	2
ADC$_\uparrow$	92	8	29	10	5	8
ADD$_\downarrow$	275	1	18	2	23	2
TD$_\downarrow$	8101	166	183	147	4123	4
STD$_\downarrow$	364933	1360	1849	3079	892305	2
TDIR$_\uparrow$	1	0.8888	0.8444	1	0.2	11
DAIR$_\uparrow$	0.1470	0.9756	0.8085	0.9574	0.6	2
WDD$_\uparrow$	-112.1999	27.8999	23.0999	34.4999	-353.6	6
Ranking$\downarrow$	5	1	2	3	4	0

Table 3: Concurrent users – Change point

proposed metrics with the performance of the classical metrics. We concluded that our metrics never gave an incorrect answer. The same could not be said of the classical methods.

We also applied our proposed metrics to two real datasets of web traffic. We compared the performance of all metrics discussed, and concluded that our proposed metrics performed better or the same as the classical metrics.

References

[1] E. Baidoo, J. Lewis Priestley,An Analysis of Accuracy using Logistic Regression and Time Series, *Grey Literature from PhD Candidates.* **2** (2016), https://digitalcommons.kennesaw.edu/dataphdgreylit/2/. ⇒115

[2] J. Caiado, N. Crato, D. Peña, A periodogram-based metric for time series classification, *Computational Statistics Data Analysis* **50** (2006) 2668–2684. ⇒115

[3] B. Esmael, A. Arnaout, R. K. Fruhwirth, G. Thonhauser, Improving time series classification using Hidden Markov Models, *Proceedings of the 12th International Conference on Hybrid Intelligent Systems (HIS)*, 2012, pp. 502–507. ⇒115

[4] A. Gensler, B. Sick, Novel Criteria to Measure Performance of Time Series Segmentation Techniques, *Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM*, Aachen, Germany, 2014. ⇒115, 120, 121

[5] R.J. Hyndman, Time Series Data Library, Accessed: 2018-11-12, https://datamarket.com/data/list/?q=provider:tsdl. ⇒126

[6] N. Laptev, S. Amizadeh, I. Flint, Generic and Scalable Framework for Automated Time-series Anomaly Detection, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1939–1947. ⇒126

[7] S.I. Lee, C.P. Adans-Dester, M. Grimaldi, A.V. Dowling, P.C. Horak, R.M. Black-Schaffer, P. Bonato, J.T. Gwin, Enabling stroke rehabilitation in home and community settings: a wearable sensor-based approach for upper-limb motor training, *IEEE journal of translational engineering in health and medicine* **6** (2018) 1–11. ⇒115

[8] Gh. Sebestyen, A. Hangan, Gy. Kovacs, Z. Czako, Platform for Anomaly Detection in Time-Series, *XXXIV. Kandó Conference*, Budapest, Hungary, 2018. ⇒126

Received: September 17, 2019 • Revised: November 26, 2019