Adv Syst Sci Appl 2017; **3**:22–33
Published online at http://ijassa.ipu.ru/ojs/ijassa/article/view/497

Conformal Kernel Expected Similarity for Anomaly Detection in Time-Series data

Aleksandr Safin[1,2], Evgeny Burnaev[2,3*]

[1] *National Research University Higher School of Economics, Moscow, Russia*
[2] *Skolkovo Institute of Science and Technology, Skolkovo, Moscow Region, Russia*
[3] *Institute for Information Transmission Problems, Moscow, Russia*

Abstract: The problem of anomaly detection arises in many practical applications. Currently it is highly important to be able to detect outliers in data streams, as recent years have seen a rapid growth in the amount of such data. Only a few techniques are applicable to real-time data and even fewer could provide an interpretable anomaly score. Probabilistic interpretation of the anomaly score could allow an analyst to choose the anomaly threshold based on the desired false alarm rate, which is highly important in a number of real-life applications. We propose a modification of the EXPoSE algorithm for anomaly detection in time series data, which produces a probabilistic score of abnormality. The proposed algorithm is developed within the framework of conformal anomaly detection and utilizes the expected similarity as a measure of non-conformity.

Keywords: anomaly detection, conformal prediction, time series, kernel methods, expected similarity

1. INTRODUCTION

There are many cases in which it is highly important to determine whether a new observation comes from the same distribution or not. This problem is referred to as outlier or anomaly detection. E.g. when a fitted model is applied to new data, it should be checked whether a test data set belongs to the same population as the training data set. To address the issue of novelty detection, anomaly detection techniques can be used. Anomaly detection has proven to be helpful for certain medical purposes, fraud detection and machine diagnostics, to name but a few. For instance, in [1] failure prediction for aircrafts is considered.

The definition of an anomaly varies between algorithms and applications. In general, an anomaly "is an element whose properties differ from the majority of the other elements under consideration which is called as *normal data*" [14]. In [15] anomaly detection is described as follows: "*Anomaly detection* refers to the problem of finding patterns in data that do not conform to expected behavior".

Summing up, the problem of anomaly detection can be formulated as follows: the task is to determine for every object in a test set whether it is a normal or abnormal instance in comparison with observations from a training set.

Anomaly detection approaches can be divided in the following three groups [15]:

- *Unsupervised* approaches use only the assumption that most observations are normal. Such assumption favours incremental and autonomous learning in data streams.

[*] Corresponding author: e.burnaev@skoltech.ru

- *Supervised* approaches require availability of a labelled training set containing instances of both normal and abnormal objects.
- *Semi-supervised* approaches require a small amount of labeled data with a large amount of unlabeled data.

In many practical cases a number of outliers is significantly smaller than a number of target observations, and thus usual classification methods may yield unsatisfactory results as classes in a dataset are very imbalanced. The significant dominance of target instances over outliers is a natural property of real-life data: e.g. in case of air traffic safety problems accidents happen very rarely. Another reason for that is the impossibility or very high costs of reproducing faulty conditions when we consider a machine diagnostic task. In the light of known and outlined difficulties, classical methods are not applicable to solving these problems, thus a variety of outlier detection methods have been developed. Such problems justify the need for specialized approaches to anomaly model selection [2], learning with privileged information [6], construction of ensembles of non-parametric anomaly detectors in data streams [3, 7, 8], usage of specific time-series models [9–13], and explicit rebalancing of normal and abnormal classes [4, 5], among others.

Unsupervised anomaly detection does not require the training dataset to be labelled, thus it is applicable to various problems, as in general it is not feasible to collect labels. Therefore a number of applications adopts unsupervised approaches, e.g. based on density estimation or clustering [16].

According to the surveys [15] and [17] unsupervised anomaly detection techniques could be generally categorized as probabilistic (distribution- and density-based), prediction-based, distance-based, classification-based, clustering-based and information-theoretic approaches.

Distribution-based methods estimate parameters of a target data distribution and determine whether a test object comes from the same distribution that generated samples from the training set. The main drawback of these methods is the necessity to select some parametric class of data distributions. One of the tricks is to model the target distribution as a mixture of Gaussians, however the number of Gaussians still have to be determined. To mitigate this problem, other non-parametric techniques could be utilized, for instance histogram-based or kernel density estimator (KDE). However, the number of bins should be firstly specified, and the performance is highly sensitive to this hyper-parameter. For multivariate problems a basic approach is to estimate a histogram per each input feature. However some features could be correlated, in that case the information about such dependency will be lost.

Prediction-based techniques predict future observations based on previous items and then compare predicted and real data to identify anomalies.

Another type of approaches is based on the distance to the k-th nearest neighbour (kNN). One of such techniques is the kNN-based outlier detector [18]. All objects are sorted w.r.t. the average distance to k nearest neighbours and top n of them with the highest average distance are claimed to be anomalies. LOF method, proposed in [19], exploits density based approach: it uses the distance to the k-th nearest neighbour as an inverse estimate of a local density value. However data could contain clusters with different densities leading to significantly increased false anomaly detections.

Main drawback of distances-based anomaly detection methods is poor interpretability of their output. To address this issue Conformal Anomaly Detector (CAD) was proposed by Laxhammar [16]. Having a probabilistic interpretation of the degree of anomalousness allows choosing a threshold with a false alarm rate guarantee. Zhao and Saligrama stated in [20] that "while [modern anomaly detection] approaches provide impressive computationally efficient solutions on real data, it is generally difficult to precisely relate tuning parameter choices to desired false alarm probability". At the same time according to Burnaev and Nazarov, conformal prediction could be used for constructing non-parametric confidence intervals with a specified confidence probability [21, 22].

Some techniques adopt approaches used for classification tasks. Tax and Duin proposed Support Vector Data Description [23] and later it was refined in [24]. The task of data description is formulated as follows: given the unlabelled training data, construct a closed boundary (or a set of them) that contains predominantly the target data, and outliers are outside this boundary. In a simple case, boundary is supposed to be spherical, but in general it is possible to determine an arbitrary-shaped flexible boundary by using *kernel functions*. Moreover, SVDD is robust against the training data containing outliers and also is capable of improving the accuracy by incorporating additional information about negative examples, in case when the training dataset is labelled. According to the results of their study, SVDD is shown to yield mostly comparable or even better results for sparse and complex multidimensional datasets. An extension of SVM to the case of unlabelled data is outlined in [25]. This approach which is referred to as one-class SVM has been adapted by Ma and Perkins [26] for time series.

The problem of anomaly detection has many dimensions. In particular, data could be not fixed but represented as a stream. A variety of techniques could be used for anomaly detection. However, only a few can be used for data streams. Classical algorithms for anomaly detection are not applicable due to their computational complexity and memory consumption, since the number of elements in the set is growing and therefore it is not possible to store all previously observed data. It is worth emphasizing that frequently the concept of anomaly could change in the course of time. This phenomenon is called as *concept drift*. In general, a streaming version of an anomaly detection algorithm should have an ability of adaptation to a concept drift; therefore, such algorithms are usually based on one of the following strategies of accumulating information about current changes in data: namely windowing and exponential smoothing, which is also referred to as decay. The above-mentioned issues are considered in the paper [14]. The proposed *EXPoSE* algorithm is developed within the framework of *reproducing kernel Hilbert space* (RKHS) and exploits the concept of *kernel mean embedding*. In a nutshell, the estimator used in the algorithm could be described as a dot product of a *kernel mean map* and a *feature map* of the observed data point. However, the produced anomaly score has a lack of intuitive interpretability and therefore the false alarm rate could not be guaranteed when anomaly threshold is chosen. In order to eliminate this drawback we utilize Lazy Drifting Conformal Detector procedure proposed in [32] to construct the algorithm with probabilistic interpretation of the anomaly score while based on the idea of kernel mean embedding proposed in [14].

The structure of the paper is the following. Sections 2 and 3 shed light on kernels and conformal anomaly detection respectively. The proposed algorithm is described in Section 4. In Section 5, the results of empirical evaluation (using Numenta Anomaly Benchmark) of the proposed algorithm are outlined. Finally, Section 6 describes achieved results and the work still to be done.

2. OVERVIEW OF KERNEL-BASED METHODS FOR ANOMALY DETECTION

In machine learning kernels are broadly used for handling data of diverse nature. Therefore it is not surprising that a number of anomaly detection methods are based on the kernel framework. In this section we give some necessary definitions and provide an overview of such anomaly detection methods which uses kernels and therefore are applicable to data of various types.

2.1. Introduction to Kernels

Reproducing kernel Hilbert Space (RKHS) is a Hilbert Space $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ of functions $f\colon \mathcal{X} \to \mathbb{R}$ if the evaluational functional $\bar{\delta}_{\mathcal{X}}\colon f \to f(\mathcal{X})$ is continuous.

Reproducing kernel of $\mathcal{H}$ is a function K: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which satisfies the reproducing property:

$$\langle f, K(\mathcal{X}, \cdot)\rangle = f(\mathcal{X}),$$
$$\langle K(\mathcal{X}, \cdot), K(\mathcal{Y}, \cdot)\rangle = K(\mathcal{X}, \mathcal{Y}).$$

The map ϕ: $\mathcal{X} \to \mathcal{H}$ with the property that $K(\mathcal{X}, \mathcal{Y}) = \langle \phi(\mathcal{X}), \phi(\mathcal{Y})\rangle$ is referred to as a *feature map*.

Definition 2.1 (Expected Similarity Estimation)**:**
The expected similarity [14] of $z \in X$ given the probability distribution $\mathbb{P}(x)$ is defined as:

$$\eta(z) = \mathbb{E}_{\mathcal{X}}[\phi(z)] = \int_{\mathcal{X}} K(z, x) d\mathbb{P}(x).$$

Definition 2.2 (Kernel embedding)**:**
Kernel embedding of the distribution $\mathbb{P}$ has the form

$$\mu[\mathbb{P}] = \int_{\mathcal{X}} K(x, \cdot) d\mathbb{P}(x).$$

Expectation of any $f \in \mathcal{H}$

$$\mathbb{E}_{\mathcal{X}}[f] = \langle f, \mu[\mathbb{P}]\rangle_{\mathcal{H}}.$$

Thus,

$$\eta(z) = \langle \phi(z), \mu[\mathbb{P}]\rangle_{\mathcal{H}}.$$

Given the empirical distribution $\mathbb{P}_n(x)$ by observing n realizations $\{x_1, \ldots x_n\}$ independently sampled from $\mathbb{P}$, one could approximate $\mu[\mathbb{P}]$ as follows:

$$\mu[\mathbb{P}] \approx \mu[\mathbb{P}_n] = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i).$$

This approach is referred to as *empirical kernel embedding* [29] and given that $\|\phi(x)\| \leq C, C > 0$ the following guarantee has been proved by Schneider [30] for all $\epsilon > 0$:

$$P(\|\mu[\mathbb{P}] - \mu[\mathbb{P}_n]\| \geq \epsilon) \leq 2e^{-\frac{n\epsilon^2}{8C^2}}.$$

Considering above mentioned, having observed $\{x_1, \ldots x_n\}$, the expected similarity estimation for $z \in X$ is

$$\eta(z) = \langle \phi(z), \mu[\mathbb{P}]\rangle_{\mathcal{H}} \approx \langle \phi(z), \mu[\mathbb{P}_n]\rangle_{\mathcal{H}} = \frac{1}{n} \sum_{i=1}^{n} K(z, x_i).$$

2.2. *EXPoSE*

EXPected Similarity Estimation (EXPoSE) that was proposed in [14] is the method for anomaly detection which could handle data streams.

For every new observation z the algorithm computes an anomaly score $\eta(z)$ based on computed empirical kernel mean map w_t of previously observed items

$$\eta(z) = \frac{\langle \phi(z), w_t\rangle}{\|w_t\|^2}.$$

Kernel mean map could be evaluated using one of the following strategies. The first strategy is to use a sliding window of length l:

$$w_t = \frac{1}{l} \sum_{i=t-l+1}^{t} \phi(x_i).$$

More flexible approach is to apply exponential smoothing to all previous observations:

$$w_t = \gamma\phi(x_t) + (1-\gamma)w_{t-1}, t > 1.$$

The parameter γ reflects the influence of a new data item. However, as already was indicated, the anomaly score provided by this algorithm could not be interpreted in a probabilistic manner, therefore we propose an approach to transform the anomaly score produced by EXPoSE. To that end, the idea of Conformal Anomaly Detection is adopted to build an anomaly detector for online data.

3. CONFORMAL ANOMALY DETECTION

Laxhammar [16] proposed a Conformal Anomaly Detection (CAD) which is a distribution-free procedure for probability-like confidence measure estimation based on non-conformity measure (NCM) provided by some detector. The NCM $A(x, y)$ reflects how different the investigated object y is from other observations x. NCM could be for instance the average distance to k neighbours, the distance to the k-th neighbour, residual in a regression model, to name just a few.

Let us consider a time series x_t, then compute scores $a_s^t = A(X_{:t}^{-s}, x_s), s = 1, \ldots, t$, where $A(x, y)$ is an NCM used by the algorithm.

Then the empirical p-value is defined as:

$$p(x_t, X_{:(t-1)}, A) = \frac{1}{t}|\{s = 1, \ldots, t : a_s^t \geq a_t^t\}|.$$

The lower it is, the lower the probability of falsely rejecting the null hypothesis (x_t is anomaly) is, thus the more likely x_t is an anomaly instance.

Shafer and Vovk proved [31] the fact that CAD could provide the following guarantee when x_t is i.i.d:

$$\mathbb{P}_{\mathbf{x}\sim D}(p(\mathbf{x}_t, X^{-t}, A) < \epsilon) \leq \epsilon, X = (\mathbf{x}_s)_{s=1}^t.$$

It is clear that CAD could be computationally heavy as it requires computations of $A(X_{:t}^{-s}, x_s)$ for s from 1 to t.

To mitigate this problem, an Inductive Conformal Anomaly Detection (ICAD) was proposed by Laxhammar and Falkman in [27]. This approach relies on scores computed on training set $\bar{X}$ for every instance of the calibration set. For further simplicity, let us consider relabelled sequence $\mathbf{x}_t$ that starts from $-n+1$. Then, ICAD has the following setup for every $t \geq 1$:

$$\underbrace{\mathbf{x}_{-n+1}, \ldots, \mathbf{x}_0}_{\bar{X} \text{ training}}, \overbrace{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{t-1}}^{\text{calibration}}, \mathbf{x}_t.$$

In that setup, the conformal p-value of a test object $\mathbf{x}_t$ is computed on the basis of modified scores:

$$\{a_s^t = A(\bar{X}, \mathbf{x}_s), s = 1, \ldots, t\},\ \bar{X} = (\mathbf{x}_{-n+1}, \ldots, \mathbf{x}_0).$$

However, by relaxing deterministic guarantee to probably approximately correct guarantee, it is achievable to adapt ICAD to use only fixed size calibration set. Offline ICAD was developed to use a calibration set only with fixed size m, sliding along the time series, as illustrated:

$$\underbrace{\mathbf{x}_{-n+1}, \dots, \mathbf{x}_0}_{\bar{X}\text{ training}}, \dots, \overbrace{\mathbf{x}_{t-m}, \mathbf{x}_{t-m+1}, \dots, \mathbf{x}_{t-1}}^{\text{calibration}}, \mathbf{x}_t.$$

It should be highlighted that the conformal p-value in this case uses a subsample of the ICAD non-conformity scores:

$$p(\mathbf{x}_t, X_{:(t-1)}, A) = \frac{1}{m+1} |\{s = 0, \dots, m : a^t_{t-s} \geq a^t_t\}|.$$

Vovk proved [28] the following guarantee for the offline ICAD:

$$\mathbb{P}_{\mathbf{x}\sim D}(p(\mathbf{x}, X, \bar{A}) < \epsilon) \leq \epsilon + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

4. PROPOSED APPROACH FOR ANOMALY DETECTION IN TIME SERIES DATA

In this section we outline the proposed algorithm for anomaly detection in time series. It is worth emphasising that the developed approach does not require any assumption about the data distribution and it outputs a probabilistic measure of anomality based on non-conformity scores. Such measure of anomality is calculated using an adaptation of ICAD to the case of potentially non-stationary and quasi-periodic time series. CAD and online ICAD are computationally complex, therefore offline ICAD seems much suitable for the task. Nevertheless, as offline ICAD uses a fixed training set, it should be noticed that one could face problems in case of non-stationary time series. In the light of the discussed details and difficulties, Lazy Drifting Conformal Detector (LDCD) has been proposed in our paper [32].

For simplicity we consider a univariate time-series $X = (x_t)_{t\geq 1} \in \mathbb{R}$, although our approach is valid for multivariate data as well since we are going to use kernel-based non-conformity measure. To begin with, the time series X should be embedded into L-dimensional space. To that end, we further consider the sequence of $\mathbf{x}_t = (x_{t-L+1}, \dots, x_t) \in \mathbb{R}^L$ constructed by moving window of the width L on the time series X:

$$\dots, x_{t-L-1}, \rlap{$\overbracket{\phantom{x_{t-L}, x_{t-L+1}, \dots, x_{t-1}}}^{\mathbf{x}_{t-1}}$} x_{t-L}, \underbracket{x_{t-L+1}, \dots, x_{t-1}, x_t}_{\mathbf{x}_t}, x_{t+1}, \dots.$$

It should be noticed that such approach obviously produce $t - L + 1$ embeddings from the sequence of the length t. In other words, to produce the first such embedding, we need to observe L instances initially.

As NCM we are using the expected similarity:

$$A(T_t, \mathbf{x}_t) = \frac{1}{n} \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_{t-m-i}),$$

where $T_t = \{\mathbf{x}_s : s = t - m - n, \dots, t - m - 1\}$, $K(x, y)$ is a kernel function.

$$\begin{array}{rlll} \text{data:} & \dots, \overbrace{\mathbf{x}_{t-m-n}, \dots, \mathbf{x}_{t-m-1}}^{T_t \text{ training}}, & \mathbf{x}_{t-m}, \dots, \mathbf{x}_{t-1}, & \overbracket{\mathbf{x}_t}^{\text{test}}, \dots \\ \text{scores:} & \dots, \mathbf{a}_{t-m-n}, \dots, \mathbf{a}_{t-m-1}, & \underbrace{a_{t-m}, \dots, a_{t-1}}_{A_t \text{ calibration}}, & \underbracket{a_t}_{\text{test}}, \dots \end{array}$$

The proposed approach could be described as follows:

1. Construct time series embedding in a sliding window,
2. Compute $a_{n+s} = A(T_{n+s}, \mathbf{x}_{n+s}), s = 1, \ldots, m+1$,
3. Evaluate the empirical p-value of its non-conformity score:

$$p(\mathbf{x}_t, T_t, A) = \frac{1}{m+1}|\{i = 0, \ldots, m : a_{t-i} \geq a_t\}|.$$

The algorithm is depicted in the Figure 4.1.

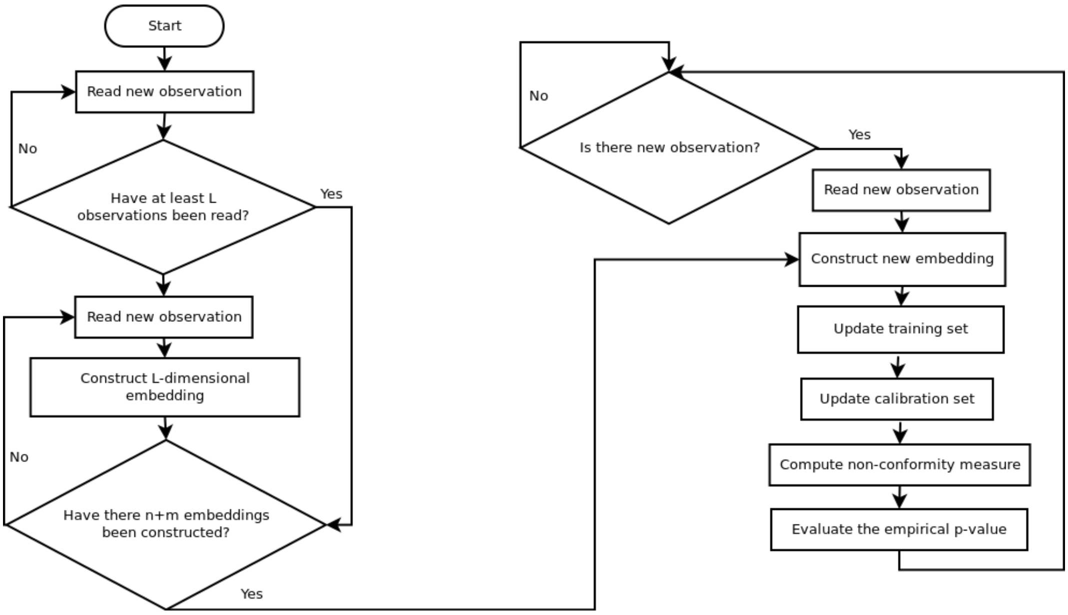
Start
Read new observation
No
Have at least L observations been read?
Yes
Read new observation
Construct L-dimensional embedding
No
Have there n+m embeddings been constructed?
Yes
No
Is there new observation?
Yes
Read new observation
Construct new embedding
Update training set
Update calibration set
Compute non-conformity measure
Evaluate the empirical p-value

Fig. 4.1. Flowchart of the algorithm

It is worth emphasising that the proposed approach could be implemented with time complexity equal to $O(n) + O(\log m)$ in the case of using red-black tree for the calibration set.

5. RESULTS ON NUMENTA ANOMALY BENCHMARK

The results of the EXPoSE LDCD comparison with several other algorithms is presented in this section along with the testing methodology. The Numenta Anomaly Benchmark (NAB) is utilized to test the proposed algorithm.

5.1. Datasets

The NAB corpus consists of 58 both real-world and artificial time series datasets. Real-world data are obtained from such sources as AWS server metrics, Twitter volume, advertisement clicking metrics, traffic data, to name just a few.

We also conduct experiments using Numenta Anomaly Benchmark on Yahoo! S5 dataset [34] which has been created to gauge the anomaly detectors performance on different types of anomalies. This corpus is divided into 4 groups: first one contains real production metrics from different Yahoo! properties, and the rest are synthetic time series.

5.2. Scoring algorithm

Commonly used metrics for performance evaluation such as accuracy, precision and recall do not suit well for anomaly detection, since they do not consider time. The NAB proposes such an approach for scoring which rewards only early true detection, meanwhile penalizes late detections and punishes false alarms very hard.

To capture early detections, NAB considers the area which is centred around the anomaly point which is referred to as *anomaly window*. The window length is defined as 10% of the length of the time series. All detections within this window are *true positives*, but only the earliest one contributes in the total score, the others will be ignored. The detections outside the anomaly window are *false positives*, missed anomalies are *false negatives*. *True negatives* are not considered in the scoring mechanism.

Bellow we described scoring scheme used in NAB [33]. An example of time series is provided in Figure 5.2. The first 15% of the time series is considered as *probationary period* and during this period an algorithm learns patterns from the data and is not required to do any detections. Then the algorithm is evaluated on the remaining part of time series. The weights for accuracy calculation is evaluated using the smooth sigmoid function as depicted in Figure 5.3.

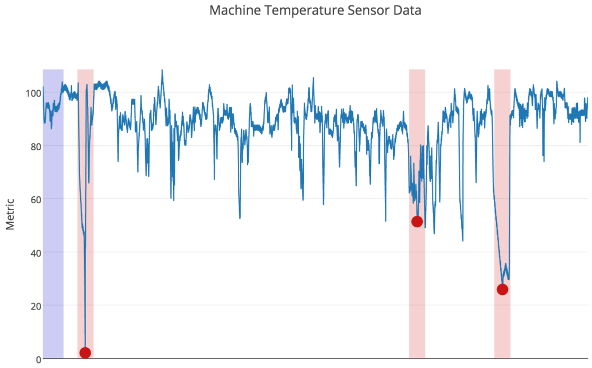

Fig. 5.2. The purple shaded area is the probationary period. Anomalies are depicted as red points and red shaded regions represent anomaly windows [33].

As the costs of true positive (TP), false positive (FP) and false negative (FN) vary among distinct applications, in NAB this is captured by an *application profile* which reflects the contribution of weights for TP, FP an FN detections.

The "Standard" application profile reflects scenarios in which misdetections have identical costs. The "Reward Low FP" and "Reward Low FN" profiles penalize harder for FP and FN respectively.

Profile	A_{TP}	A_{FN}	A_{FP}	A_{TN}
Standard	1.0	-1.0	-0.11	1.0
Reward Low FP	1.0	-1.0	-0.22	1.0
Reward Low FN	1.0	-2.0	-0.11	1.0

Table 5.1. The detection rewards on NAB application profiles

The reward for the detection depends on the relative position t of the alarm (about possible anomaly) to the left side of the anomaly window:

$$\sigma^A(t) = (A_{TP} - A_{FP}) \left(\frac{1}{1 + e^{5t}} \right) - 1.$$

The raw performance score on the dataset X with respect to application profile A is the sum of the scores over all detections plus the impact of missed anomalies (false negatives) captured by the number of anomaly windows with no detections f_{det}:

$$S^A_{det}(X) = \sum_{y \in Y_{det}} \sigma^A(y) + A_{FN} f_{det}.$$

The overall performance of the algorithm is the sum of raw performance scores over the all datasets D: $S^A_{det} = \sum_{X \in D} S^A_{det}(X)$. The final normalized performance score is determined by:

$$S^A_{NAB} = 100 \frac{S^A_{det} - S^A_{null}}{S^A_{perfect} - S^A_{null}}.$$

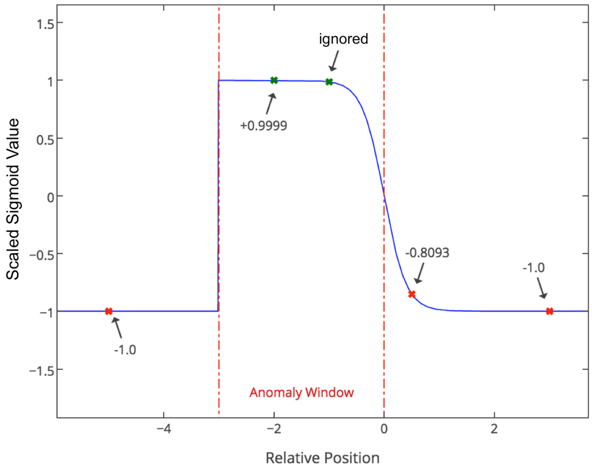

Fig. 5.3. NAB weighted scores: detections outside the anomaly window are false positives and punished; only earliest detection inside window is true positive and it will be counted, other will be ignored [33].

5.3. *Results*

Since proposed EXPoSE LDCD is conservative and demonstrates high level of false alarms, we have applied the following simple pruning strategy to reduce the false alarm rate: we output $1 - p$ as anomaly score for the observation x_t and if p is greater than 99.65%, then output of the detector is fixed at 0.5 for the next $\frac{n}{5}$ observations (n is the length of probationary period). The proposed approach has been validated on both the Numenta Anomaly Benchmark corpus and the Yahoo! S5 dataset. Tables 5.2 and 5.3 reflect the results of the algorithms comparison.

5.4. *Automated kernel bandwidth tuning*

Kernel-based methods are sensitive to the choice of bandwidth, therefore we modify the algorithm to choose the bandwidth of the kernel based on the best value of the bandwidth for Kernel Density Estimator obtained by 3-fold cross-validation. The proposed modification demonstrates significantly better results on NAB dataset and is able to increase the score on both *Low FN* and *Low FP* profiles, meanwhile it results in slight score decrease on *Standard* profile.

6. CONCLUSION

In this paper we propose an algorithm for anomaly detection in time series data, utilizing the concept of expected similarity and applying framework of conformal anomaly detection.

Table 5.2. Results on Numenta Anomaly Benchmark

Detector \ Profile	Standard	Reward Low FP	Reward Low FN
Numenta HTM	70.1	63.1	74.3
EXPoSE LDCD +tuning	45.53	25.77	54.78
Windowed Gaussian	39.6	20.9	47.4
EXPoSE LDCD	37.93	20.14	45.11
Etsy Skyline	35.7	27.1	44.5
Bayesian Changepoint	17.7	3.2	32.2
EXPoSE	16.4	3.2	26.9

Table 5.3. Results on Yahoo! S5 dataset

Detector \ Profile	Standard	Reward Low FP	Reward Low FN
EXPoSE LDCD	51.88	38.76	58.95
EXPoSE LDCD +tuning	49.79	43.73	61.45
Numenta HTM	41.0	37.5	44.4
Bayesian Changepoint	35.7	17.6	43.6
EXPoSE	32.09	7.00	45.45
Windowed Gaussian	31.1	25.8	40.7
Etsy Skyline	23.6	18.0	28.9

Table 5.4. Average running time performance on NAB dataset

Detector \ Performance	items per second	ms per item
Windowed Gaussian	1984.862	0.504
EXPoSE LDCD	1500.224	0.667
Bayesian Changepoint	428.639	2.333
EXPoSE	398.496	2.51
Numenta HTM	98.012	10.202
Etsy Skyline	4.582	218.229

Table 5.5. Average running time performance on Yahoo! S5 dataset

Detector \ Performance	items per second	ms per item
EXPoSE LDCD	2548.293	0.392
Windowed Gaussian	2348.041	0.426
Bayesian Changepoint	1217.888	0.821
EXPoSE	383.711	2.606
Numenta HTM	103.777	9.636
Etsy Skyline	4.656	214.773

This approach has been rigorously validated on NAB corpus and Yahoo! S5 dataset using Numenta Anomaly Benchmark. On both datasets the proposed approach excel the EXPoSE, which produces expected similarity as anomaly score and expected similarity is used as non-conformity measure in the LDCD procedure. Moreover, the developed algorithm shows great running time performance, which is important for online detectors and achieves high results on standard profile on Yahoo dataset. Also, the implementation of the algorithm could be enhanced, as it has not been thoroughly optimised and it could be one of the directions for future research. We also propose a tuning procedure for the kernel bandwidth parameter, however there is still a significant room for improvements.

ACKNOWLEDGEMENTS

The work was supported by The Ministry of Education and Science of Russian Federation, grant No. 14.606.21.0004, grant code: RFMEFI60617X0004.

References

1. Alestra S., Bordry C., Brand C., Burnaev E., Erofeev P., Papanov A. & Silveira-Freixo C. (2014) Application of Rare Event Anticipation Techniques to Aircraft Health Management *Advanced Materials Research*, **1016**, 413–417.
2. Burnaev E., Erofeev P. & Smolyakov D. (2015) Model Selection for Anomaly Detection. *Proc. SPIE9875, Eighth International Conference on Machine Vision (ICMV 2015)*, **987525**, http://dx.doi.org/10.1117/12.2228794
3. Artemov A. & Burnaev E. (2015) Ensembles of Detectors for Online Detection of Transient Changes. *Proc. SPIE9875, Eighth International Conference on Machine Vision (ICMV 2015)*, **98751Z**, http://dx.doi.org/10.1117/12.2228369
4. Burnaev E., Erofeev P. & Papanov A. (2015) Influence of Resampling on Accuracy of Imbalanced Classification. *Proc. SPIE9875, Eighth International Conference on Machine Vision (ICMV 2015)*, **987521**, http://dx.doi.org/10.1117/12.2228523
5. Burnaev E., Erofeev P. & Papanov A. (2017) Meta-learning for Construction of Resampling Recommendation Systems. *ArXiv e-prints*, 1706.02289, [Online]. Available:https://arxiv.org/abs/1706.02289
6. Burnaev E & Smolyakov D. (2016) One-Class SVM with Privileged Information and Its Application to Malware Detection. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 273–280.
7. Burnaev E., Ishimtsev V., Bernstein A. & Nazarov A. (2017) Conformal k-NN Anomaly Detector for Univariate Data Streams. *Proceedings of Machine Learning Research*, **60**, 213–227.
8. Volkhonsky D., Burnaev E., Nouretdinov I., Gammerman A. & Vovk V. (2017) Inductive Conformal Martingales for Change-Point Detection. *Proceedings of Machine Learning Research*, **60**, 132–153.
9. Artemov A. & Burnaev E. (2016) Optimal sequential estimation of a signal, observed in a fractional gaussian noise. *Theory of Probability and Its Applications*, **60**(1), 126–134.
10. Artemov A. & Burnaev E. 2016) Detecting Performance Degradation of Software-Intensive Systems in the Presence of Trends and Long-Range Dependence. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 29–36.
11. Artemov A., Burnaev E. & Lokot A. (2015) Nonparametric Decomposition of Quasi-periodic Time Series for Change-point Detection. *Proc. SPIE 9875, Eighth International Conference on Machine Vision*, **987520**.
12. Burnaev E. (2009) Disorder Problem for Poisson Process in Generalized Bayesian Setting. *Theory Probab. Appl.*, **53**(3), 500–518.
13. Burnaev E., Feinberg E. & Shiryaev A. (2009) On Asymptotic Optimality of the Second Order in the Minimax Quickest Detection Problem of Drift Change for Brownian Motion. *Theory Probab. Appl.*, **53**(3), 519–536.
14. Schneider M., Ertel W. & Ramos Fabio T. (2016) Expected Similarity Estimation for Large-Scale Batch and Streaming Anomaly Detection. *Machine Learning*, **105**(3), 305–333, https://doi.org/10.1007/s10994-016-5567-7
15. Chandola V., Banerjee A. & Kumar V. (2009) Anomaly Detection: A Survey. *ACM Comput. Surv.*, **41**(3), 15:1–15:58.
16. Laxhammar R. (2014) Conformal anomaly detection. Detecting abnormal trajectories in surveillance applications. *Ph. D. Thesis*. University of Skövde, Skövde. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-8762
17. Pimentel M. A. F., Clifton D. A., Clifton L. & Tarassenko L. (2014) Review: A Review of Novelty Detection. *Signal Process*, **99**, 215–249.
18. Ramaswamy S., Rastogi R. & Shim K. (2000) Efficient Algorithms for Mining Outliers from Large Data Sets. *SIGMOD Rec.*, **29**(2), 427–438.
19. Breunig M.M., Kriegel H.-P., Ng R. T. & Sander J. (2000) LOF: Identifying Density-based Local Outliers. *SIGMOD Rec.*, **29**(2), 93–104.

20. Zhao M. & Saligrama V. (2009) Anomaly Detection with Score functions based on Nearest Neighbor Graphs. *NIPS'09 Proceedings of the 22nd International Conference on Neural Information Processing Systems*, Vancouver, Canada, 2250-2258.
21. Burnaev E. & Nazarov I. (2016) Conformalized Kernel Ridge Regression. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 45–52, https://doi.org/10.1109/ICMLA.2016.0017
22. Burnaev E. & Vovk V. (2014) Efficiency of conformalized ridge regression. *Proceedings of the Twenty Seventh Annual Conference on Learning Theory. JMLR: Workshop and Conference Proceedings*, **35**, 605–622.
23. Tax D.M.J. & Duin R.P.W. (2004) Support Vector Data Description. *Machine Learning*, **54** (1), 45–66.
24. Chang W.-C., Lee C.-P. & Lin C.-Jen. (2013) A revisit to support vector data description (SVDD). *Documents in the CiteSeerx database* [Online]. Available: http://ai2-s2-pdfs.s3.amazonaws.com/a244/422ba339713d0c9eaa153b378e9f9fc08263.pdf
25. Schölkopf B., Platt J.C., Shawe-Taylor J. C. et al. (2001) Estimating the Support of a High-Dimensional Distribution. *Neural Comput.*, **13** (7), 1443–1471.
26. Ma J. & Perkins S. (2003) Time-series novelty detection using one-class support vector machines. *Proceedings of the International Joint Conference on Neural Networks, 2003*, **3**, 1741–1745.
27. Laxhammar R. & Falkman G. (2015) Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*, **74** (1), 67–94.
28. Vovk V. (2012) Conditional validity of inductive conformal predictors. *Proceedings of the Asian Conference on Machine Learning, in PMLR*, **25**, 475-490.
29. Smola A., Gretton A., Song L. & Schölkopf B. (2007) A Hilbert Space Embedding for Distributions. *Algorithmic Learning Theory: 18th International Conference, ALT 2007*, 13–31, https://doi.org/10.1007/978-3-540-75225-7_5
30. Schneider M. (2016) Probability Inequalities for Kernel Embeddings in Sampling without Replacement. *Proceedings of Machine Learning Research*, 66–74.
31. Shafer G. & Vovk V. (2008) A tutorial on conformal prediction. *J. Mach. Learn. Res.*, **9**, 371–421
32. Ishimtsev V., Nazarov I., Bernstein A. & Burnaev E. (2017) Conformal k-NN Anomaly Detector for Univariate Data Streams. *ArXiv e-prints*, [Online]. Available: https://arxiv.org/abs/1706.03412
33. Lavin A. & Ahmad S. (2015) Evaluating Real-time Anomaly Detection Algorithms - the Numenta Anomaly Benchmark. *14th International Conference on Machine Learning and Applications (IEEE ICMLA)*, 38-44, https://arxiv.org/abs/1510.03336
34. Yahoo! Webscope (2017, December 26) *S5 - A Labeled Anomaly Detection Dataset, version 1.0.* [Online]. Available https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70.