sciendo

JAISCR, 2022, Vol. 12, No. 1, pp. 19– 27
10.2478/jaiscr-2022-0002

ANOMALY PATTERN DETECTION IN STREAMING DATA BASED ON THE TRANSFORMATION TO MULTIPLE BINARY-VALUED DATA STREAMS

Taegong Kim and Cheong Hee Park*

Department of Computer Science and Engineering, Chungnam National University,
220 Gung-dong, Yuseong-gu
Daejeon, 305-763, Korea

**E-mail: cheonghee@cnu.ac.kr*

Submitted: 10th July 2020; Accepted: 6th October 2020

Abstract

Anomaly pattern detection in a data stream aims to detect a time point where outliers begin to occur abnormally. Recently, a method for anomaly pattern detection has been proposed based on binary classification for outliers and statistical tests in the data stream of binary labels of normal or an outlier. It showed that an anomaly pattern can be detected accurately even when outlier detection performance is relatively low. However, since the anomaly pattern detection method is based on the binary classification for outliers, most well-known outlier detection methods, with the output of real-valued outlier scores, can not be used directly. In this paper, we propose an anomaly pattern detection method in a data stream using the transformation to multiple binary-valued data streams from real-valued outlier scores. By using three outlier detection methods, Isolation Forest(IF), Autoencoder-based outlier detection, and Local outlier factor(LOF), the proposed anomaly pattern detection method is tested using artificial and real data sets. The experimental results show that anomaly pattern detection using Isolation Forest gives the best performance.

Keywords: anomaly pattern detection, multiple binary-valued streams, outlier detection, outlier score.

1 Introduction

An outlier detection method predicts whether a data sample is an outlier [1]. On the other hand, anomaly pattern detection in a data stream aims to find a time point where outliers suddenly begin to occur heavily. A sudden increase of outliers might indicate that an unusual event has happened [2]. The anomaly pattern detection method which was recently proposed in [3] utilizes an outlier detection method that performs the binary classification for outliers. By applying the outlier detection method to each data sample in a data stream, a data stream is transformed into the stream of binary values indicating normal or an outlier for each data sample. Then the occurrence of an anomaly pattern is detected on the stream of binary values by comparing binomial distributions in a reference window and a detection window.

Well-known outlier detection methods such as Isolation Forest(IF) [4], Autoencoder-based outlier detection [5, 6], and Local outlier factor(LOF) [7] compute outlier scores which measure the degree of anomaly in a data sample. Since the anomaly pattern detection method in [3] requires a binary out-

lier detection method in order to transform a data stream to a stream of binary values, most outlier detection methods, with the output real-valued outlier scores, can not be used directly. Although binary labels of normal or outliers can be obtained by setting a threshold over outlier scores, it is difficult to determine the optimal threshold.

In this paper, we present an anomaly pattern detection method in a data stream based on the transformation into multiple binary-valued data streams. Given a training set of normal data samples, an outlier detection model is constructed using the training set and it is applied to each data sample on a data steam, resulting in a stream of real-valued outlier scores. Multiple thresholds over outlier scores by which abnormal data samples could be distinguished from normal data samples are induced from outlier scores of normal training data samples. By applying the thresholds to a data stream of the real-valued outlier scores, multiple data streams of binary values are obtained. Two approaches for anomaly pattern detection on multiple binary-valued data streams are suggested. The proposed methods, APD-HT/IF, APD-HT/AE, and APD-HT/LOF based on Isolation Forest(IF), autoencoder-based outlier detection, and Local outlier factor(LOF)-based outlier detection methods respectively are tested comparing the performance with the method APD-HT in [3].

The contribution of the paper can be summarized as follows.

- Given normal training data, an anomaly pattern detection method on a data stream is proposed extending the method in [3].
- By setting multiple thresholds from outlier scores of training data samples, an ensemble of anomaly pattern detectors is constructed.
- Well-known outlier detection methods which give the output of outlier scores can be utilized in the proposed method.

The remainder of the paper is organized as follows. In Section 2, the anomaly pattern detection method in [3] is reviewed. In Section 3, we propose a method for anomaly pattern detection in a stream of real-valued outlier scores. Experimental results are given in Section 4 and discussion follows in Section 5.

2 Anomaly pattern detection based on binary classification of outliers

While enormous research for outlier detection has been conducted, anomaly pattern detection in a data stream has not been studied extensively [8, 2]. For anomaly pattern detection, in [9, 10], a change in probability distribution on a data stream of real-valued outlier scores was detected based on a one-dimensional Gaussian distribution assumption. In [11, 12], anomaly pattern detection was performed through rule induction on categorical attributes. The method [3] which was published recently showed competent performance for anomaly pattern detection. It detects a burst occurrence of outliers on a binary-valued data stream which is induced by binary classification for outliers.

In [3], given a training set consisting of normal data samples, an outlier detection model is constructed which gives a binary label indicating normal or an outlier for a data sample. A clustering-based outlier detection method was used for binary outlier prediction. The normal data region in the training set is modeled as a union of hyperspheres by performing k-means clustering on the training data. By partitioning training data to several chunks and applying k-means clustering for each chunk, the ensemble of cluster models can be constructed. When a test data sample is not included in the nearest hypersphere in any of the cluster models, it is predicted to be an outlier. By performing outlier prediction for each data sample in a data stream, a data stream is transformed into the stream of binary values where 1 stands for an outlier and 0 for normal.

Two methods for detecting a time point where a sudden burst of outliers occurs were proposed: APD-HT(Anomaly Pattern Detection by Hypothesis Testing) and APD-CC(Anomaly Pattern Detection by Control Charts). Since it was shown that the performance of APD-HT is generally higher than APD-CC in [3], we focus on the review of APD-HT. In APD-HT, a reference window is set in the beginning part of the binary-valued data stream which is considered as consisting of normal data samples and a detection window is moved forward as a new data sample arrives. Let X and Y be the number of outliers in the reference window and the detection

window whose size is m and n respectively. Supposing that the samples in the reference window are generated from the binomial distribution of the proportion p_1 and the samples in the detection window are generated from the binomial distribution of the proportion p_2, a hypothesis test about the equality of proportions is performed. Under the null hypothesis $H_0 : p_1 - p_2 = 0$ and alternative hypothesis $H_1 : p_1 - p_2 < 0$,

$$p\text{-value} = P(Z < z_0),$$
$$\text{where } \hat{p}_1 = \frac{X}{m}, \hat{p}_2 = \frac{Y}{n}, \hat{p} = (X+Y)/(m+n),$$
$$\text{and } z_0 = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})(1/m+1/n)}},$$

is computed. For a significance level α, if the p-value is less than α, then H_0 is rejected and anomaly pattern detection is declared. Otherwise, a detection window moves forward for a new hypothesis test.

3 Anomaly pattern detection based on the transformation to multiple binary-valued data streams

Binary classification of outliers based on k-means clustering performs reasonably well and the anomaly detection method in [3] is less affected by the performance of outlier prediction. However, if a well-known outlier detection method can be used, it will improve anomaly pattern detection as well as outlier pattern detection.

Given training data consisting of normal data samples, an outlier detection model is constructed which computes an outlier score of a data sample. From the distribution in outlier scores of training data samples, a threshold for outlier prediction can be induced. However, it is difficult to expect that one threshold value optimally distinguishes outliers from normal data.

Instead of choosing one threshold for outlier prediction, we can use multiple thresholds. For each threshold, the data stream of the real-valued outlier scores is transformed into a stream of binary values indicating outlier prediction with 1 and normal prediction with 0, and the anomaly pattern detection method, APD-HT, is applied in each binary-valued data stream. Now there are two questions to answer: how to set multiple thresholds, and how to combine the results from APD-HT on multiple binary-valued data streams. In the next subsections, these problems are addressed.

3.1 How to set multiple thresholds for outlier prediction

Suppose that outlier scores are small when the anomalous degree of data samples is high, as in the implementation of Isolation Forest algorithm in scikit Learn [13]. Then it is generally expected that the score of outliers is smaller than that of normal data samples. Since a training set contains only normal data samples, thresholds can be chosen using outlier scores of normal training data. We choose the threshold using the percentile numbers in the distribution of outlier scores of normal training data samples. A percentile is a value below which a given percentage of observations in a group of observations falls. When the outlier scores of normal data samples and outliers in a test data set are expected to be separated well, threshold values smaller than 1th-percentile can work well. However, when the outlier scores of normal data samples and outliers in a test set are mixed in a wide range, using too small percentile values as a threshold can not differentiate outliers from normal data samples. In the experiments in Section 4, we test the effects of using various thresholds on the performance of anomaly pattern detection.

3.2 How to combine the results from multiple binary data streams

The anomaly pattern detection method, APD-HT, is applied to each binary data stream. A detection window moves forward while the reference window is fixed at the front of the data stream. The distributions in two windows are compared to detect the burst of 1's in the detection window. Hence, a positive or negative prediction is made in each binary data stream and these predictions are combined to make a final decision. We tested two approaches for combining the predictions from multiple binary data streams: All-Agreed and Half-Agreed approaches. In the All-Agreed approach, when the predictions from all the data streams are

positive, anomaly pattern detection is declared. On the other hand, in the Half-Agreed approach, when the majority of predictions are positive, anomaly pattern detection is declared. Table 1 summarizes the algorithm of the proposed method.

3.3 Outlier detection methods

Most of outlier detection methods calculate outlier scores which represent the degree at which a data sample deviates from the normal data range. Among various outlier detection methods, we used Isolation Forest and Autoencoder-based outlier detection and Local outlier factor(LOF)-based outlier detection in the experiments of Section 4.

Isolation Forest(IF) [4] is a well-known tree-based outlier detection method. Isolation Forest is based on the assumption that outliers are easy to isolate from the remainders of the data. It grows a binary tree by selecting randomly an attribute and a splitting value of the attribute. The process is repeated recursively until all training data samples are isolated at the leaf nodes. The set of isolation trees built on subsets of a training data is called isolation forest. The length of a path where a data sample traverses from the root node to a leaf node is used to compute an outlier score.

Recently, various deep learning-based methods have been used for outlier detection. Autoencoder consists of two components of neural networks. An input data is encoded to low dimensional representation by the first network called an encoder, and it is again decoded to the original dimensional space by the second network called a decoder. Autoencoder is trained so that the error between the reconstructed one and the input data is minimized. Also, the error is used to compute an outlier score.

LOF(Local outlier factor) gives an outlier score based on local density around a data sample [14].

$$LOF(p) = \frac{\frac{1}{k}\sum_{q \in \mathrm{kNN}(p)} lrd(q)}{lrd(p)} \qquad (1)$$

$\mathrm{kNN}(p)$ denotes a set of k nearest neighbors of a data sample p. Local reachability density(lrd) of p is computed from the inverse of the average reachability distance to the k nearest neighbors of p. LOF provides an indication of whether p is in a denser or sparser region of the neighborhood than its neighbors [15].

Table 2. Data description

Data	attr.	samples	outlier ratio(%)
Creditcard	28	284,807	0.17
Kdd-Http	3	567,498	0.39
Annthyroid	6	7,200	7.42
Shuttle	9	49,097	7.15
Gaussian	1	102,500	2.44
RBFevents	5	100,000	10.2
Covtype	10	286,048	0.96
Satellite	36	5,100	1.4
Mammography	6	11,183	2.3

4 Experiments

4.1 Experimental Setup

To compare the performance of the proposed anomaly pattern detection method, we used nine data sets including real or artificial data. Detailed description is shown in Table 2. Creditcard data[1] is a summary of credit card usage by some card holders in Europe in September 2013. Excluding the attribute indicating usage amount, 28 attributes were used. KDD-http data is a subset of KDD Cup data[2], which is composed of data samples whose value of attribute *service* was http. Three attributes, *duration*, *src-bytes*, and *dst-bytes*, were used as in [16]. Gaussian data is 1-dimensional data where 100,000 normal samples were generated using 5 Gaussian mixture distributions with mean values 0, 1, 2, 3 and 4, and 2,500 abnormal data samples were generated from Gaussian distribution with the mean value 6. RBFevents data was constructed using the artificial streaming data generator RandomRBFevents of MOA [17] with normal data from five normal distributions and outliers from a random uniform distribution.

Other data sets were downloaded from the OpenML data repository. In Annthyroid data, the data samples in two classes, *hyperfunction* and *subnormal functioning*, were considered to be outliers. In Shuttle data, data samples of class 1 were treated

[1] https://www.kaggle.com/dalpozz/creditcardfraud

[2] kddcup.data-10-percent-corrected data by the DARPA Intrusion Detection Evaluation Program: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Table 1. The algorithm of the proposed method.

Input: X: a training set of normal data samples
$x_1, x_2, x_3, \cdots$: a test data stream
$\{p_1, p_2, \cdots, p_k\}$: a set of threshold values
Combining strategy: All-Agreed or Half-Agreed

Construct the outlier detection model F by using X.
Apply F for data samples in X and obtain a set of outlier scores G.
Let p_ith-percentile in G be s_i. $(i = 1, \cdots, k)$
Initialize k empty binary streams $B_1, \cdots, B_k$.
for $t = 1, 2, 3, \cdots$
  Compute the outlier score for x_t by applying F.
  The binary value obtained by the threshold s_i is added to the end of the binary stream B_i for $i = 1, \cdots, k$.
  Apply APD-HT on each binary stream B_i $(i = 1, \cdots, k)$, where a reference window is fixed
    in the beginning part of a data stream and a detection window is set at the end of the stream
    including the newly arrived instance.
  Combine the predictions from all k binary streams by the combining strategy.
  if anomaly pattern detection is signaled
    Give an alarm for anomaly pattern detection and exit
  end if
end for

Table 3. Comparison of anomaly pattern detection performance. Thresholds of 0.1th, 0.5th, 1th, 2th, and 3th percentiles were used.

	APD-HT [3]	APD-HT/IF		APD-HT/AE		APD-HT/LOF	
F1		All	Half	All	Half	All	Half
Creditcard	1	1	0.84	1	0.8	0	0.84
Kdd-Http	1	1	0.79	1	0.8	0	0.02
Annthyroid	0.96	0.96	0.98	1	0.95	1	0.98
Shuttle	0.84	1	0.93	0.6	0.91	1	0.91
Gaussian	1	1	0.9	1	0.88	0.99	0.84
RBFevents	1	1	0.92	1	0.9	1	0.87
Covtype	1	1	0.87	1	0.86	1	0.84
Satellite	0.96	0.98	0.96	1	0.99	1	0.98
Mammo.	0.92	1	0.98	1	0.98	0.04	0.99
mean	0.96	**0.99**	0.91	0.96	0.9	0.67	0.81
	APD-HT [3]	APD-HT/IF		APD-HT/AE		APD-HT/LOF	
Delay		All	Half	All	Half	All	Half
Creditcard	13.2	20.6	15.6	22.8	15.9	-	75.5
Kdd-Http	27.2	15.2	10.8	15.2	10.6	-	11
Annthyroid	72.7	129.6	43.6	15.8	9.4	67.7	32.9
Shuttle	7.6	15.6	9.5	215.9	12.1	15.8	9.4
Gaussian	5.9	17.1	11.1	15.4	9.6	18.8	11.8
RBFevents	5.0	14.7	8.9	14.7	9.6	16.3	10.8
Covtype	134.3	183.8	30.5	15.0	9.8	17.2	12.3
Satellite	21.9	101.3	32.9	14.5	9.4	22.7	15.8
Mammo.	50.0	20.5	14.7	16.1	10.2	210	42.7
mean	37.5	57.6	19.7	38.4	10.7	52.6	24.7

as normal data and data samples in the other classes except class 4 were set as outliers as in [4]. Covtype data targets the prediction of forest cover types from cartographic variables. 10 numeric attributes were used and data samples of class 2 were treated as normal data and data samples of class 4 were set as outliers. In Shuttle data, data samples of class *cotton crop* and *soil with vegetable stubble* were used as outliers.

Each data sample is split to a training set which consists of normal data corresponding to 30% of the total data samples and a test set of remaining data samples by which a test data stream is built. In order to simulate the occurrence of an anomaly pattern on a test data stream, outliers were arranged after a sequence of normal data, and the starting point of outliers was set as the actual anomaly pattern occurrence point. An outlier detection model is constructed by the training data and anomaly pattern detection is performed on a test data stream.

The experiment was repeated 100 times by randomly splitting to training and test data. For each test case, when anomaly pattern occurrence is predicted after the actual anomaly pattern occurrence point, it is counted as TP (True Positive) prediction. When an anomaly pattern is predicted before the actual anomaly pattern occurrence point, it is considered FP(False Positive) prediction. If anomaly pattern occurrence is not detected until the end of the test stream, it is marked as FN (False Negative) prediction. In a case of a true positive detection, a distance from the actual anomaly pattern occurrence point to the anomaly pattern prediction point is measured as *Delay*. After 100 experiments, from the accumulated TP, FP, and FN, the F1 value is computed by Equation 2 along with the average value of *Delay*.

$$F1 = \frac{2 * precision * recall}{precision + recall}, \tag{2}$$

where $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$.

4.2 Comparison of anomaly pattern detection performance

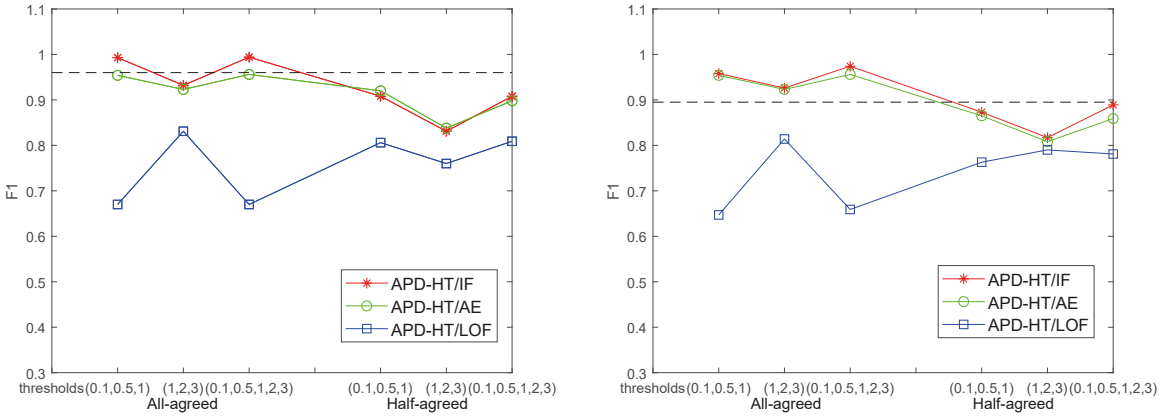
The performance of the proposed anomaly pattern detection method was compared with the method APD-HT in [3]. APD-HT [3] uses the ensemble of k-means clustering for binary outlier prediction. As in [3], the ensemble of three clustering models with the number of clusters 30 was used. The proposed methods, APD-HT/IF, APD-HT/AE, and APD-HT/LOF, are based on outlier detection methods of Isolation Forest(IF), Autoencoder(AE), and Local outlier factor(LOF), respectively. For Isolation Forest, the implementation in scikit-learn [13] was used, where the sub-sampling size was the number of training data samples and the ensemble size was 100. The tree height limit H is automatically set by the sub-sampling size ψ as $H = ceiling(log_2\psi)$. Outlier scores by Autoencoder and LOF were computed using the implementation in PyOD which is a Python toolkit for outlier detection [18]. In Autoencoder, 6 hidden layers were set where the number of nodes in an encoder part increased 1.5 times per layer and decreased reversely per layer in a decoder part. Default parameter values in PyOD implementation were used where relu activation function was utilized in all the hidden layers and sigmoid function in the output layer, and Adam optimizer was used. In LOF, the number of neighbors was set to 20 which is a default value in PyOD, and the Euclidean distance metric was used. For Autoencoder and LOF methods, preprocessing of the standard normalization was performed for all the data sets except Gaussian data which is one-dimensional data.

For all compared methods, the size of the reference window and the detection window was set as 400, and the significance level of the hypothesis test was set at 0.01 as in [3]. Five thresholds of 0.1th, 0.5th, 1th, 2th, 3th percentiles were set for the transformation from the stream of outlier scores to binary-valued streams. Table 3 shows the performance of the compared methods in 100 repeated tests by F1 and the average value of Delay. Isolation Forest outlier detection method combined with APD-HT showed the highest F1 value compared with other methods. The all-agreed strategy gave higher F1 value in APD-HT/IF and APD-HT/AE than the half-agreed approach, while the average delay is smaller when using the half-agreed approach.

As in [3], we also tested anomaly pattern detection performance in noisy environments. This was simulated by inserting 10% outliers randomly into the normal data sequence and mixing the same

Table 4. Comparison of anomaly pattern detection performance in the simulation of a noisy environment. Thresholds of 0.1th, 0.5th, 1th, 2th, 3th percentiles were used in the proposed methods.

	APD-HT [3]	APD-HT/IF		APD-HT/AE		APD-HT/LOF	
F1		All	Half	All	Half	All	Half
Creditcard	1	1	0.85	1	0.84	0	0.85
Kdd-Http	1	0.98	0.86	1	0.77	0	0
Annthyroid	0.95	0.95	0.96	0.99	0.96	1	0.96
Shuttle	0.74	0.96	0.87	0.57	0.82	0.95	0.84
Gaussian	0.84	0.98	0.85	0.97	0.84	1	0.86
RBFevents	0.68	0.91	0.8	0.92	0.8	0.94	0.78
Covtype	1	1	0.83	0.99	0.8	0.99	0.77
Satellite	0.94	0.98	0.98	1	0.98	1	0.98
Mammo.	0.91	1	0.99	1	0.92	0.04	0.98
mean	0.90	**0.97**	0.89	0.94	0.86	0.66	0.78
	APD-HT [3]	APD-HT/IF		APD-HT/AE		APD-HT/LOF	
Delay		All	Half	All	Half	All	Half
Creditcard	14.9	22.2	16.5	27.2	17.8	-	72.8
Kdd-Http	30.5	17.1	11.6	15.9	11.9	-	-
Annthyroid	83.3	138.5	53.3	18.7	15.7	86.6	37.0
Shuttle	12.7	18.7	13.6	377.3	17.6	21.8	15.4
Gaussian	9.3	19	14.5	18.4	13.1	20.7	16.1
RBFevents	15.9	22.2	17.6	20.8	18.0	22.9	17.0
Covtype	135.2	198.2	33.7	16.7	11.5	20.9	14.7
Satellite	27.2	105.6	39.1	16.5	11.3	24.4	18.0
Mammo.	55.5	24.2	17.4	20.1	13.5	233	48.7
mean	42.7	62.9	24.1	59.1	14.5	61.5	30.0



(a) In the experimental setting of Table 3 (b) In the simulation of a noisy environment of Table 4

Figure 1. Comparison of anomaly pattern detection performance when different threshold values were used.

number of normal data with the outlier data sequence. Table 4 compares the performance in the simulated noisy situations. APD-HT/IF showed the decrease from 0.99 to 0.97 in the F1 value on average compared with the performance in non-noisy situation of Table 3. On the other hand, in APD-HT [3] the decrease from 0.96 to 0.9 in the F1 value was obtained.

4.3 Performance comparison under various threshold values

Figure 1 compares the average F1 values in nine data sets when different threshold values were used in the experimental setting of Table 3 and 4. With All-agreed strategy, APT-HT/IF and APT-HT/AE obtained high F1 values when relatively small threshold values such as 0.1th-percentile and 0.5th-percentile were included in the set of thresholds. On the other hand, APT-HT/LOF shows a different behavior pattern with APT-HT/IF and APT-HT/AE. It is conjectured that when outlier scores do not well describe the outlier degree of data samples, it is not easy to determine thresholds from the outlier scores of normal training data samples.

5 Discussions

In this paper, we proposed an anomaly detection method in a data stream that utilizes outlier scores by a well-known outlier detection method. When a new data sample arrives, the outlier score for the data sample is computed by the outlier detection model which is constructed from normal training data. Then binary values obtained by applying multiple thresholds for the outlier score are added to the end of the binary-valued data stream corresponding to each threshold. The thresholds are computed from percentile values on the outlier scores of normal training data samples. The binomial distribution in a detection window that moves forward on a binary data stream is compared with the distribution on the reference window at the beginning part of the binary data stream. The predictions made in multiple binary data streams are combined to make a decision for anomaly pattern detection. If anomaly pattern occurrence is detected, an alarm signal is given. Otherwise, the process is repeated for the new incoming data sample.

Among the three outlier detection methods, IF, Autoencoder, and LOF, the anomaly pattern detection using IF, APD-HT/IF, demonstrated the best detection performance on average. It can detect the occurrence of anomalous events by focusing on the pattern where outliers are predicted rather than the accuracy in outlier prediction for an individual data sample. As future work, we intend to apply the proposed anomaly detection method in a real application area such as breakdown detection in production facilities or new topic detection in a social network.

Acknowledgments

This work was supported by research fund of Chungnam National University.

References

[1] D. Hawkins. Identification of outliers. Springer Netherlands, 1980.

[2] C.H. Park. Outlier and anomaly pattern detection on data streams. The Journal of Supercomputing, 75:6118–6128, 2019.

[3] T. Kim and C.H. Park. Anomaly pattern detection for streaming data. Expert Systems with Applications, 149, 2020.

[4] F. Liu, K. Ting, and Z. Zhou. Isolation forest. In Proceedings of the 8th International Conference on Data Mining, 2008.

[5] Q. Feng, Y. Zhang, C. Li, Z. Dou, and J. Wang. Anomaly detection of spectrum in wireless communication via deep auto-encoders. The Journal of Supercomputing, 73(7):3161–3178, 2017.

[6] P. Remy. Anomaly detection in time setries using auto encoders. bolg positng from http://philipperemy.github.io/anomaly-detection.

[7] D. Pokrajac, A. Lazarevic, and L.J. Latecki. Incremental local outlier detection for data streams. In Proceedings of the CIDM, 2007.

[8] C. Aggarwal. Outlier analysis. Springer, 2017.

[9] D. Padilla, R. Brinkworth, and M. McDonnell. Performance of a hierarchical temporal memory network in noisy sequence learning. In Proceedings of IEEE international conference on computational intelligence and cybernetics, 2013.

[10] S. Ahmad and S. Purdy. Real-time anomaly detection for streaming analytics, 2016. Retrieved from https://arxiv.org/pdf/1607.02480.pdf.

[11] W. Wong, A. Moore, G. Cooper, and M. Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In Proceedings of the 18th International Conference on Artificial Intelligence, 2002.

[12] K. Das, J. Schneider, and D. Neil. Anomaly pattern detection in categorical datasets. In Proceedings of KDD, 2008.

[13] F. et al. Pedregosa. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[14] M.M. Breunig, H-P. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In Proceedings of the 2000 ACM Sigmod International Conference on Management of Data, 2000.

[15] P. Tan, M. Steinbach, and V. Kumar. Introduction to data mining. Addison Wesley, Boston, 2006.

[16] S. Hawkins, H. Hongxing, G. Williams, and R. Baxter. Outlier detection using replicator neural networks. In Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, 2002.

[17] A. Bife, G. Holmes, R. Kirkby, and B. Pfahringer. Moa: Massive online analysis. Journal of Machine Learning Research, 11:1601–1604, 2010.

[18] Y. Zhao, Z. Nasrullah, and Z. Li. Pyod: A python toolbox for scalable outlier detection. Journal of Machine Learning Research, 20(96):1–7, 2019.



Taegong Kim received his M.S. in Computer Science from Chungnam National University, Korea in 2020. He is currently working at Research Institute, SmartPro, Korea. His research interests include machine learning, deep learning, data mining.



Cheong Hee Park received her Ph.D. in Mathematics from Yonsei University, Korea in 1998. She received her M.S. and Ph.D. degrees in Computer Science from University of Minnesota, USA in 2002 and 2004, respectively. She has been on faculty at Department of Computer Science and Engineering, Chungnam National University, Korea since 2005, where currently she is a professor. Her research interests include machine learning, pattern recognition, data mining.