Scale Variable Dynamic Stream Anomaly Detection with Multi-chain Queue Structure

Fei Wu[1(✉)], Ting Li[1], Kongming Guo[2], and Wei Zhou[2]

[1] State Grid Fujian Electric Power Company, Fuzhou 350001, China
[2] State Grid Info-Telecom Great Power Science and Technology Co., Ltd., Fuzhou 350003, China

Abstract. Most existing data stream anomaly detection algorithms do not involve in the tackling of multi-scale characteristic of stream data, a multi-chain queue based data storage structure that is especially suitable for the analysis of multi-scale stream data is designed, and a corresponding algorithm to identify the multi-scale stream anomaly is proposed. The algorithm employs an iteration strategy and takes 3θ as the criteria for discriminating the anomaly, to minimize each anomaly's effect to its neighbors, to detect simultaneously the anomalies in the data sequences that are at the same time and the anomalies in different times. Meanwhile, the increase of a new data sample and the delete of an obsolete observation data are implemented effectively through the operation of a queue, Hence, a better result of stream data anomaly mining is obtained with the changing of mining scale. Finally, through the experiments on a real stream dataset, the proposed algorithm is shown to be capable of finding out some true anomalies in different scales with a higher accuracy rate, when compared with the traditional sliding window based algorithms and the machine learning based algorithms.

Keywords: Anomaly detection · Stream data · Multi-chain queue · Multi-scale

1 Introduction

Streaming data is one of the common data types in the cloud environment of the Internet of things. Exploring the algorithm, significance and application of stream anomaly detection is a hot research topic in the field of data mining in recent years [1]. On the basis of traditional outlier detection, researchers have proposed many excellent outlier detection algorithms, such as distance based method [2], sliding window based method [3] and so on. However, for the multi-scale characteristics of network flow data, there are few related studies in the literature, and there is no report on the results of multi-scale flow anomaly detection.

Hence, this paper bases on the storage structure, similarity measurement and scale aggregation strategy of multi-scale network flow data, and designs a multi-chain queue based data storage structure to conduct dynamic stream anomaly detection with the changing of scale. It is found that the algorithm is capable of finding out some true anomalies at the same time and in different times during the dynamic changing of mining scales.

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 516–525, 2021.
https://doi.org/10.1007/978-3-030-89814-4_37

2 Related Work

Stream anomaly detection algorithm and its application is a hot research field of data mining in recent ten years. In terms of algorithm mechanism, related research mainly includes methods based on statistics and sliding window, methods based on data mining and methods based on artificial intelligence.

In the area of sliding window, Kontaki et al. [2] studied anomaly detection based on distance, and proposed an anomaly recognition method for continuous monitoring data stream based on sliding window. Zhang et al. [3] proposed an angle based subspace anomaly detection method based on sliding window. Lin et al. [4] studied anomaly detection of data flow in sensor networks based on sliding window and optimized clustering. Qiu et al. [5] proposed a stream data anomaly detection method based on long-term memory (LSTM) network and sliding window, aiming at the characteristics of large amount of stream data and rapid production. It can not only predict data, but also update and adjust the network in real time while learning. Yu et al. [6] proposed a data flow anomaly detection algorithm based on angle variance to solve the problem of high-dimensional spatial sparsity, and demonstrated its application in elevator fault detection.

In the aspect of data mining, Salehi et al. [7] studied the incremental local outlier detection algorithm to save memory, and proposed an efficient flow anomaly detection algorithm MiLOF. Gao et al. [8] studied the incremental stream data outlier mining based on cube. Kim et al. [9] used binary classification strategy for statistical testing of anomalies, thus realizing anomaly pattern detection of data flow. Zhu et al. [10] proposed an improved approximate average KNN outlier detection scheme based on grid for IOT flow data, which uses grid to filter most normal data, and also makes the data anomaly test simple. Ma et al. [11] proposed a network abnormal traffic recognition method based on bag of words model clustering to solve the problems of low recognition accuracy of existing flow anomaly detection methods and the need to determine the threshold for rapid recognition.

In the field of artificial intelligence, Bouguelia et al. [12] proposed an incremental unsupervised flow anomaly detection algorithm GNG-A with the object of data flow undergoing different types of changes and the goal of algorithm adaptive update maintenance. Based on neural network and Bayesian optimization, Alnafessah et al. [13] carried out hybrid anomaly detection of data stream. Xu et al. [14] proposed an abnormal network data mining algorithm based on fuzzy neural network for the influence of fuzzy weighted disturbance on network data clustering center.

Although researchers have carried out a lot of fruitful research in the aspects of algorithm strategy, adaptive ability, parallelizability, real-time, high-dimensional complexity, concept drift and so on, and also put forward some theoretical or practical methods, there is no report on the related results of multi-scale flow anomaly detection in the literature.

To sum up, many researchers have carried out a lot of meaningful research on the strategy, adaptive ability, parallelism of flow anomaly detection algorithm, as well as the real-time performance, high-dimensional sparsity, concept drift of flow data, and also put forward some very valuable methods. However, there are few related reports on the research of multi-scale flow anomaly, so the paper focuses on multi-scale flow anomaly detection, especially the design and implementation of flow anomaly detection algorithm are explored.

3 Scale Variable Dynamic Stream Anomaly Detection

In order to achieve efficient multi-scale anomaly detection of dynamic network flow data, one of the core tasks is to design the storage structure of flow data and control the storage space of flow data within a certain range. On the other hand, to design a scientific multi-scale aggregation scheme and its corresponding definition of similarity measurement. Therefore, the multi-scale dynamic flow anomaly detection algorithm based on composite chain queue is introduced from the aspects of data storage structure, multi-scale aggregation function, similarity measurement and core algorithm.

3.1 Multi-chain Queue Structure

In view of the dynamic multi-scale characteristics of network flow data, the flow data storage structure of composite chain queue type shown in Fig. 1 is designed. Stream data is stored in different chain queue sequences according to scale conditions that is noted as s, which can only be processed once and the sequence is guaranteed. At the same time, it is suitable for multi-scale stream anomaly analysis.

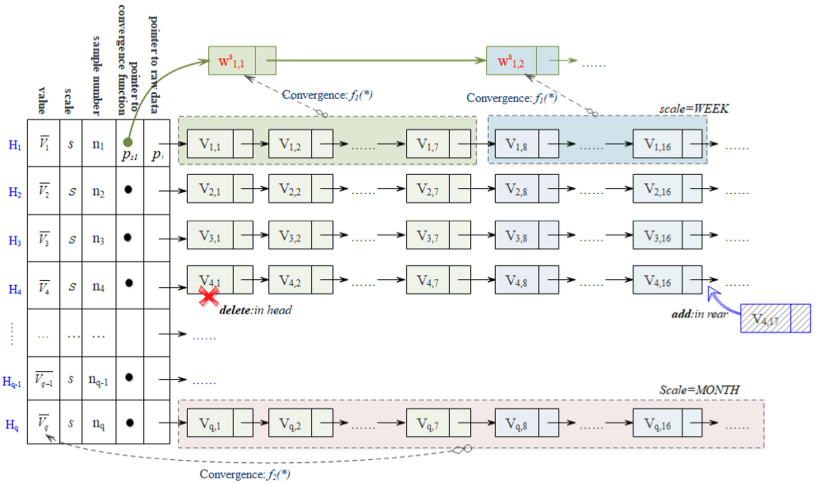

Fig. 1. The storage of network stream data in multi-queue structure.

In the chain queue structure, when a new stream data sample is added, the sample is stored in the rear of the specific chain queue according to the time $H_x(x = 1, 2, ..., q)$ and the scale condition s, such as $V_{4,1}$ in Fig. 1; while an observation is deprecated, the oldest sample in the head of the chain queue is deleted, such as $V_{4,17}$ in Fig. 1. In addition, the pointer p_x in the chain header node points to the list nodes, where the raw stream data observation values $V_{x,y}$ are stored, while x is the time of data acquisition, for 24 h in a day, $x = 24$, and is the index number of the data sample.$V_{x,y}$. in series … In order to distinguish the direction of data flow, data flow..$V_{x,y}$.. is subdivided into input flow and output flow, that is $V_{x,y} = (IS, OS)_{V(x,y)}$, denotes the input flow data value and

OS denotes the output flow data value. Furthermore, the pointer p_{sx} in the chain header node points to a series of data value $w^s_{x,z}$, which are aggregated from the raw stream data $V_{x,y}$ under the scale condition s specified in the header node of the chain queue. The aggregation function f_1 is usually the average value, while.z. is the serial number of the data sample for the moment x under the scale condition s, $z \in \{1, 2, 3, ...\}$. The aggregation data also distinguishes the input stream from the output stream.

3.2 Multiscale Convergence Function

When the network flow data is converged according to the scale condition s, the aggregation function f_1 used is shown in Eq. (1), that is, the k_{th} observation value. $w^s_{x,k}$. at the moment x with the scale condition s is generated by the function f_1. Similarly, the aggregation value $\overline{V_x}$ with the scale condition s is converged from $w^s_{x,k}$, by the convergence function f_2 (as shown in Eq. (2)), that is, $\overline{V_x}$ is generated by the convergence of the observation sequence $w^s_{x,k}$, $k = 1, 2, 3, ...$

$$w^s_{x,k} = f_1(x_k) = \frac{1}{\|s\|} \cdot \sum_{i=1}^{s} v_{x,\,(k-1)s+i} \tag{1}$$

$$\overline{V_x} = f_2(w^s_x) = \frac{1}{|n_x|} \cdot \sum_{k=1}^{n_x} w^s_{x,k} \tag{2}$$

3.3 Dissimilarity Measurement

As the core element of anomaly detection, dissimilarity measurement between data samples is defined by weighted Euclidean distance for stream data including incoming and outgoing flow, as shown in Eq. (3).

$$diff\,(V_1, V_2) = \omega \cdot (IS_{V_1} - IS_{V_2}) + (1 - \omega) \cdot (OS_{V_1} - OS_{V_2}) \tag{3}$$

Wherein, ω is the weight of incoming and outgoing flow. For the server under attack, it is better to take a higher ω value; for the server carrying out program instruction broadcast or data transmission, it is better to take a lower ω value.

3.4 Stream Anomaly Detection Algorithm

In the multi-chain queue stream data storage structure as shown in Fig. 1, when the scale condition s is changed, the stream data sequence $w^s_{x,z}$ $(x = 1, 2, 3, ..., q)$ with scale s is obtained through the aggregation function f_1 on the basis of raw stream data observation $V_{x,y}$, then, the overall stream value $\overline{V_x}$ for the moment x is computed through the aggregation function f_2 on the basis of $w^s_{x,z}$. Obviously, the stream data sequence $w^s_{x,z}$ is the original stream data observation sequence, when s is the original granularity of stream data. Therefore, the proposed stream anomaly detection algorithm should carry out anomaly detection on each series of stream data sequence $w^s_{x,z}$ $(z = 1, 2, 3, ...)$, as

well as on the sequence of overall stream value $\overline{V_x}$ ($x = 1, 2, 3, ..., q$), to find out the outlier in each series and in each time moment. Finally, when the scale condition s is changed, each data sequence $w_{x,z}^{s}$ and the overall stream value $\overline{V_x}$ need to be updated and re-aggregated, and the stream anomaly detection need to be carried out again.

For the stream data series $P = \{P_1, P_2, \cdots, P_n\}$, the algorithm employs the 3θ criterion (θ is the standard deviation of the stream data observation in a sequence) to judge if an observation is abnormal. Moreover, the algorithm uses an iterative strategy to reduce the influence of a significant deviation outlier on the other data observation, that is, only the most abnormal data is detected in each cycle, and the program loops until all anomalies are detected.

According to the strategy of the algorithm, the pseudo-code for the iterative stream anomaly detection algorithm is as follows.

Set ***findIreOutlier***($P = \{P_1, P_2, \mathrm{L}\ , P_n\}$, *threshold*=3θ){
(1) $S = \varnothing$;
(2) while(TRUE){
(2.1) $C_S = \varnothing$;
(2.2) $\overline{P} = compute\_Mean(P)$, $\theta = compute\_Std(P)$;
(2.3) for(i=1; i<=n; i++){
$value = diff(P_i, \overline{P})$;
if(value > 3θ) $C_S = C_S \cup P_i$;
(2.4) if($C_S = \varnothing$) break;
else{ $P_O = find\_max\{C_S\}$; $S = S \cup P_O$; $P_O = \overline{P}$ }
}
(3) return S ;
}

Step (1) initializes the final anomaly set S to be empty $\emptyset$; Step (2) loops circulates until no more abnormality was found, while sub-step (2.1) initializes candidate anomaly set C_S to be empty $\emptyset$, sub-step (2.2) calculates the mean $\overline{P}$ and standard deviation θ of data series P with the function $\overline{P} = compute\_Mean(P)$ and $\theta = compute\_Std(P)$ respectively, sub-step (2.3) computes the dissimilarity $diff(P_i, \overline{P})$ between P_i and for each data sample P_i by using Eq. (3). If $diff(P_i, \overline{P})$ is greater than 3θ, P_i is regarded as a candidate anomaly and is added to.C_S.. Finally, sub-step (2.4) obtains the data object P_O with the largest deviation value (that is $P_O = \max\{C_S\}$), to be the anomaly found in current cycle, and P_O is added to S, and replayed with the $\overline{P}$ to eliminate the impact of the anomaly P_O on the other data.

3.5 Efficiency Analysis

For the data sequence with n objects, step (2.1) is finished in constant time, step (2.2) is to compute the mean and standard deviation whose time complexity is $O(n)$, step (2.3) is to calculate the deviation of each data sample whose time complexity is $O(n)$, and step (2.4) is to select the data with the largest deviation from the candidate anomalies,

and the time complexity is not more than $O(n)$. Therefore, for the data sequence P with m exceptions, the time complexity of the algorithm is $O(n \cdot m)$.

For the multi-chain queue based multi-scale stream anomaly detection algorithm shown in Fig. 1, the time complexity of the stream anomaly is $O(q \cdot m)$, where m is the number of anomalies and q is the series number of chain queues. Among which, the time complexity for the detection of series $w^s_{x,z}$ $(H_x,\ x = 1, 2, 3, ..., q)$ with scale condition s is $O(t_x \cdot m_x)$, where t_x is the scale of the data sequence $w^s_{x,z}$ and m_x is the number of anomalies of $w^s_{x,z}$. Hence, the time complexity of anomaly detection is $O(t_x \cdot m_x \cdot q)$ for a total number of q data sequence. Therefore, the overall time complexity of multi-scale stream anomaly detection algorithm is $O(q \cdot m) + O(t_x \cdot m_x \cdot q)$.

4 Experiments

4.1 Data Information

The experimental is conducted on a PC with Windows 7, Intel Core i5-3570 CPU @ 3.4 GHz × 2, 8 GB memory. The object is the number of data packets obtained by four application servers for network supervision. The time span is from January 2010 to February 2011. The total number of data is 27234, including ID, machine number and eight characteristics of data capture, such as year, month, day, hour, minute, second and flow rate. The number of valid data of four servers A1, A2, A3 and A4 is 8077807581002857 respectively.

The five characteristics for the network flow data obtained from the four servers are shown in Table 1.

Table 1. Five characteristics of the data set

	Min	Q1	Q2	Mean	Q3	Max
A1	0	0	0	1.956	4.539	12.788
A2	0	5.616	9.234	104.559	12.969	52718.629
A3	0	1.384	1.451	397.335	1203.25	1699.000
A4	0	11.775	13.344	12.436	14.252	18.354

It can be seen from Table 1 that the median (Q2), the third quantile (Q3) and maximum (max) of servers A1, A2 and A3 are significantly different, which means that there is a large fluctuation in network flow and the possibility of abnormality is high; while the characteristic value of server A4 is relatively flat and there is no obvious fluctuation. Therefore, in order to highlight the value and advantages of multi-scale flow anomaly detection, the multi-scale anomaly detection of server A4 is introduced and the results are analyzed.

4.2 Result and Discussion

According to the 3θ criterion, the data from server A4 is iteratively detected on the "hour" scale, and the results are shown in Fig. 2. A total of 12 iterations were carried out, and 329 anomalies were detected (green labeled points shown in Fig. 2). The upper and lower thresholds and the number of anomalies of each round of detection are shown in Fig. 3.

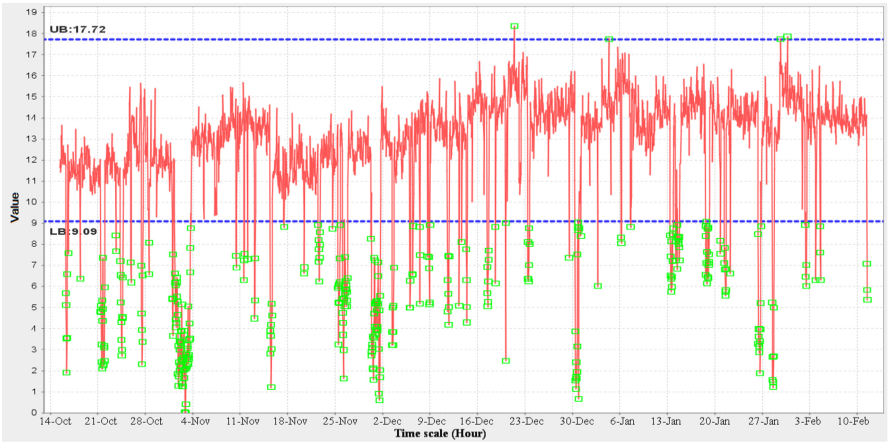

Fig. 2. Anomaly detection result with $s =$ Hour, $IR = 12$.

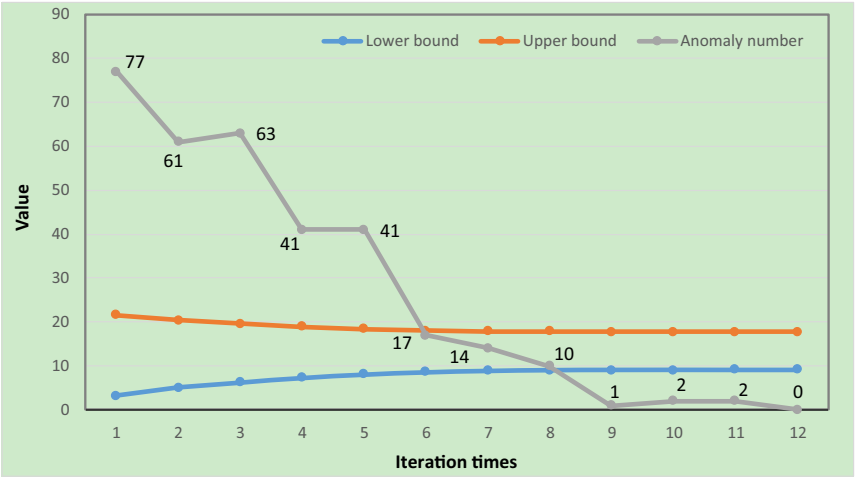

Fig. 3. Anomaly number with different upper and lower bound thresholds when $s =$ Hour.

On the "day" scale, the iterative anomaly detection is performed for each time (24:00) sequence according to "day". The number of iterations and the number of anomalies are shown in Fig. 4. Compared with the "hour" and "day" scale, the number of anomalies (128) is significantly reduced, and the number of iterations is also greatly reduced, which is in line with the high granularity contraction characteristics of flow anomalies.

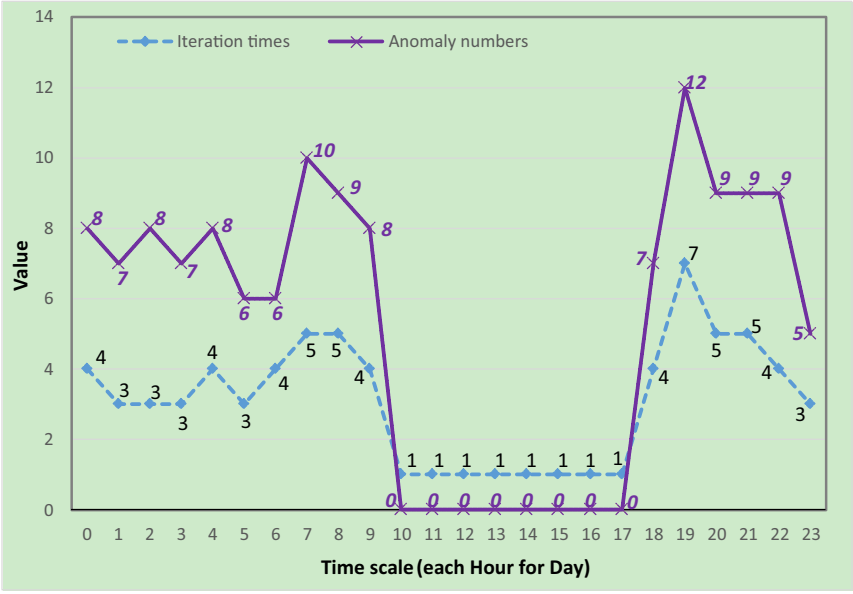

Fig. 4. Iteration times and anomaly numbers for different series of each time when s = Day.

On the "day" scale, iterative anomaly detection is performed on the corresponding sequence of each time (24 h) by "week". The number of iterations and the total number of anomalies are shown in Fig. 5.

Comparing with Fig. 4 and Fig. 5, it can be seen that with the rolling up of time scale, some non-significant anomalies are diluted, and the number of anomalies and the number of iterative detection are correspondingly reduced, such as 1–3 and 20–23 periods. In addition, under different scale conditions, although the number of anomalies is different, but the trend is roughly the same, so that people can basically determine whether there is an anomaly at a certain time or period, for example, there is an anomaly at 8 and 18, and there is less possibility of an anomaly at 11–17.

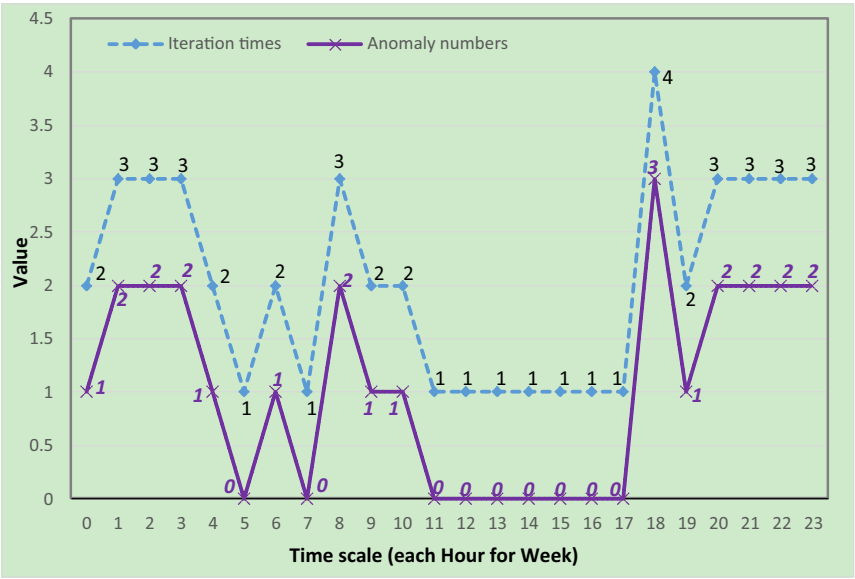

Fig. 5. Iteration times and anomaly numbers for different series of each time when s = Week.

According to the above analysis, mining and determining abnormal data on multiple time scales can obtain more reliable results and meet the needs of practical application.

5 Conclusions

This paper designs a multi-chain queue structure based algorithm for the detection of multi-scale dynamic stream anomaly. The algorithm can detect simultaneously the anomalies in the data sequence at a certain time and at each time under a given scale, and maintain the ability of anomaly detection when the scale conditions change, to realize the detection of multi-scale dynamic flow anomalies. The algorithm guarantees the sequence of stream data through the composite chain queue, and only processes it once, while the increase of new data and the elimination of historical data are efficiently realized through the queue in and out operations. In addition, the algorithm controls the memory usage by limiting the size of the composite chain queue storage structure, so as to ensure the ability of the algorithm to process massive stream data. Experimental research and efficiency analysis show that the proposed algorithm can detect anomalies in different scales, and then obtain more meaningful anomalies in multi-scale.

Further research work includes the control of total amount flow data storage in queued, and the anomaly measurement of high-dimensional flow data, and the reduction of false detection rate.

Acknowledgement. The work was supported in part by the Natural Science Foundation of Fujian Province, China (No. 2020H0043) and the Science and Technology Project of State Grid Fujian Electric Power Company (No. 5213001800LF).

References

1. Wang, H., Bah, M.J., Hammad, M.: Progress in outlier detection techniques: a survey. IEEE Access **7**, 107964–108000 (2019)
2. Kontaki, M., Gounaris, A., Papadopoulos, A.N., Tsichlas, K., Manolopoulos, Y.: Efficient and flexible algorithms for monitoring distance-based outliers over data streams. Inf. Syst. **55**, 37–53 (2016)
3. Zhang, L., Lin, J., Karim, R.: Sliding window-based fault detection from high-dimensional data streams. IEEE Trans. Syst. Man Cybern. Syst. **47**(2), 289–303 (2017)
4. Lin, L., Su, J.: Anomaly detection method for sensor network data streams based on sliding window sampling and optimized clustering. Safety Sci. **118**, 70–75 (2019)
5. Qiu, Y., Chang, X., Qiu, Q., Peng, C., Su, S.: Stream data anomaly detection method based on long short-term memory network and sliding window. J. Comput. Appl. **40**(05), 1335–1339 (2020)
6. Yu, L., Li, Y., Zhu, S.: Anomaly detection algorithm based on high-dimensional data stream. Comput. Eng. **44**(01), 51–55 (2018)
7. Salehi, M., Leckie, C., Bezdek, J.C., Vaithianathan, T., Zhang, X.: Fast memory efficient local outlier detection in data streams. IEEE Trans. Knowl. Data Eng. **28**(12), 3246–3260 (2016)
8. Gao, J., Ji, W., Zhang, L., Li, A., Wang, Y., et al.: Cube-based incremental outlier detection for streaming computing. Inf. Sci. **517**, 361–376 (2020)
9. Kim, T., Park, C.H.: Anomaly pattern detection for streaming data. Expert Syst. Appl. **149**, 1–16 (2020)
10. Zhu, R., Ji, X., Yu, D., Tan, Z., Zhao, L., et al.: KNN-based approximate outlier detection algorithm over IoT streaming data. IEEE Access **8**, 42749–42759 (2020)
11. Ma, L., Wan, L., Ma, S., Yang, T.: Abnormal traffic identification method based on bag of words model clustering. Comput. Eng. **43**(05), 204–209 (2017)
12. Bouguelia, M.-R., Nowaczyk, S., Payberah, A.H.: An adaptive algorithm for anomaly and novelty detection in evolving data streams. Data Min. Knowl. Discov. **32**(6), 1597–1633 (2018)
13. Alnafessah, A., Casale, G.: TRACK-plus: optimizing artificial neural networks for hybrid anomaly detection in data streaming systems. IEEE Access **8**, 146613–146626 (2020)
14. Xu, L., Wang, J.: Data mining algorithm of abnormal network based on fuzzy neural network. Comput. Sci. **46**(04), 73–76 (2019)