



AnomalyDetect: An Online Distance-Based Anomaly Detection Algorithm

Wunjun Huo¹, Wei Wang^{1,2(✉)}, and Wen Li¹

¹ Department of Computer Science and Engineering, Tongji University,
Shanghai, China

willtongji@gmail.com

² School of Data Science and Engineering, East China Normal University,
Shanghai, China

Abstract. Anomaly detection is a key challenge in data mining, which refers to finding patterns in data that do not conform to expected behavior. It has a wide range of applications in many fields as diverse as finance, medicine, industry, and the Internet. In particular, intelligent operation has made great progress in recent years and has an urgent need for this technology. In this paper, we study the problem of anomaly detection in the context of intelligent operation and find the practical need for high-accuracy, online and universal anomaly detection algorithms in time series database. Based on the existing algorithms, we propose an innovative online distance-based anomaly detection algorithm. K-means and time-space trade-off mechanism are used to reduce the time complexity. Through the experiments on Yahoo! Web-scope S5 dataset we show that our algorithm can detect anomalies accurately. The comparative study of several anomaly detectors verifies the effectiveness and generality of the proposed algorithm.

Keywords: Anomaly detection · Time series · Online algorithm · Euclidean distance · Intelligent operation

1 Introduction

Time series refers to a set of random variables arranged in chronological order. We can record the value of a random event in time order to obtain a time series sequence. Time series data appears in every aspect of today's life as diverse as finance, medicine, industry, and the Internet. For example, time series can be the daily settlement price of each stock, the monthly deposit balance of a customer, and the number of heart beats per minute of a patient. Finding the rules among these time series has great application prospects. In general, the research on time series includes prediction, pattern mining, clustering, classification, anomaly detection, etc. In the financial field, time series prediction can be applied in stock price forecasting. Prediction and classification can help banks to determine customer credit rating. In the medical field, pattern mining and anomaly detection can help doctors find abnormal situations in patient monitoring data quickly and make timely process.

In intelligent operation, a series of time series data can be obtained by real-time monitoring and recording on various hardware and software indicators. Some abnormal behaviors such as hardware and software failures, malicious attacks, etc. are directly reflected in these time series databases, forming abnormal data. One of the duties of the operation staff is to monitor these indicators in real time, and make timely repair when abnormalities occur. The key to intelligent operation is using anomaly detection algorithms to monitor the abnormal situation in the time series database of operation field automatically. Therefore, the detection of abnormal data in time series data becomes a key technology in intelligent operation. In this context, we analyze the existing algorithms, combine the characteristics and requirements of intelligent operation, and propose an innovative online anomaly detection algorithm.

Anomaly detection in time series refers to finding a point or a sequence, which do not conform to their expected behavior. In real cases, the abnormal points or sequences generally represent abnormal situations: illegal transaction, abnormal heartbeat, cyber malicious attacks, etc. There are many algorithms for anomaly detection in time series. The idea of most algorithms is to find the laws of normal time series data, when the current observation point or sequence is too different from the normal law, it will be regarded as abnormal.

However, in the field of intelligent operation [27], the data generated from hardware and software is streaming data. Hardware and software have a great variety of indicators, so the amount of data is large. What's more, the time series fluctuates frequently and irregularly. When the abnormal behavior occurs, the abnormal types of these data are also different. Therefore, combined with the characteristics of the operation data and the requirements of the operation scene, an appropriate anomaly detection algorithm is of great significance to the landing of intelligent operation.

The online distance-based anomaly detection algorithm proposed in this paper has several advantages to meet the requirements of intelligent operation. First, it applies the sliding window mechanism to calculate the anomaly of each point online. Second, although the orders of magnitude of different time series may be of large difference, the distributing fitting method can use the quantiles to determine the anomalies efficiently. Third, online algorithms often require a fast detection method, we take an innovative mechanism in this paper to reduce time complexity.

The rest of this paper is organized as follows: in Sect. 2, we give the main background definitions used in our proposed algorithm and review the related academic work. Section 3 gives the total steps and details of the innovative online anomaly detection algorithm proposed in this paper. Section 4 shows the performance result of the algorithm on the specific datasets. We give a summary and outlook in Sect. 5.

2 Problem Description and Related Work

This section specifies the relevant background knowledge of the anomaly detection algorithm, describes the specific problems to be solved in this paper, and introduces the related research work.

2.1 Definition and Description

A time series generally refers to a sequence in which a random variable is arranged in chronological order. The time series can be defined as follows.

Definition 1. Time Series. We use a set of random variables arranged in chronological order, i.e. $X_1, X_2, \dots, X_t, \dots$, to represent the time series of a random event, $\{X_t, t \in T\}$ in short. We use x_1, x_2, \dots, x_n or $\{x_t, t = 1, 2, \dots, n\}$ to represent the ordered observations of the random sequence, called the observation sequence with length n .

An abnormal point often refers to a data point that differs greatly from the surrounding points in a time series. An abnormal sequence refers to several continuous points in a period behave differently with their surrounding sequences. Figure 1 shows a true operation time series, which contains both abnormal points and abnormal sequences. Operation staff in real operation scene tag the anomalies, and in Fig. 1, these anomalies are marked in red.

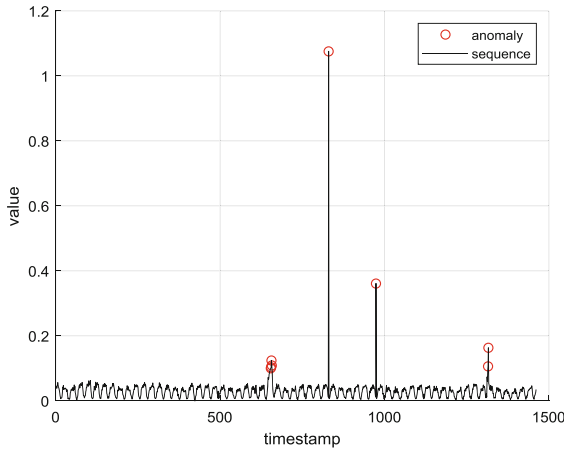


Fig. 1. The real operation data containing the abnormal data (Color figure online)

Figure 1 shows that the abnormal points and abnormal sequences in time series usually have large differences with the values or morphologies of their surrounding points. Therefore, when determining whether a point belongs to an abnormal point, it is necessary to consider the relevant points around it. For the determination of anomalies, the normal sequence model needs to be compared as a sample. Therefore, the definition of time subsequence and sliding window is introduced in this paper as follows.

Definition 2. Subsequence. Given a time series with length n , i.e. $X = \{x_t, t = 1, 2, \dots, n\}$, a subsequence $S_{i,l}$ in time series is defined as a set of continuous points with length $l \leq n$, which is expanded forward from x_i , i.e. $S_{i,j} = \{x_{i-l+1}, x_{i-l+2}, \dots, x_{i-1}, x_i\}$, where $l \leq i \leq n$.

Definition 3. Sliding window. Given a time series X with length n , and a subsequence $S_{i,l}$ with length l , a sliding window $W_{i,m}$ in X is defined as a set of continuous points

with length $m(l \leq m \leq n)$, which is expanded forward from x_i and includes $S_{i,l}$, that is $W_{i,m} = \{x_{i-m+1}, x_{i-m+2}, \dots, x_{i-l}, x_{i-l+1}, x_{i-l+2}, \dots, x_{i-1}, x_i\}$, where $m \leq i \leq n$.

The first problem we need to solve is to detect the abnormal points in a period accurately. For the subsequence that needs to be judged whether abnormal or not, it is necessary to determine whether the subsequence is similar to the rest normal subsequences. In order to avoid Comparison, we introduce the definition of non-self-match.

Definition 4. Non-self-match. Given a time series X of length n , a subsequence $S_{i,l}$ and a sliding window $W_{i,m}$, a non-self-match of $S_{i,l}$ is defined as the subsequence $S_{j,l}$ of length l in $W_{i,m}$ which is expanded forward from x_i , that is $S_{j,l} = \{x_{j-l+1}, x_{j-l+2}, \dots, x_{j-1}, x_j\} (i - m + l \leq j \leq i - l)$.

The definition of non-self-match makes sure that the two subsequences in the given sliding window have no intersection points. When measuring the similarity between two subsequences in a sliding window, we should use a distance-based metric. In mathematics, the distance is defined as follows.

Definition 5. Distance. Let S be a non-empty set, for any two elements x and y in S , Distance is a function $d(x, y)$ that maps these two points to a real number and satisfies the following three axioms:

- i. Non-negative: $d(x, y) \geq 0$, $d(x, y) = 0$ iff $x = y$;
- ii. Symmetry: $d(x, y) = d(y, x)$;
- iii. Triangle Inequality: $d(x, y) \leq d(x, z) + d(y, z)$ for any $x, y, z \in S$.

The shape-based distance calculation methods in time series include Euclidean distance, cosine distance, and Pearson correlation coefficient [18], etc. According to the characteristic of our proposed algorithm, Euclidean distance metric is the most appropriate, and we confirm it in Sect. 2.2. Here we give the definition of Euclidean Distance.

Definition 6. Euclidean Distance. Given a time series X of length n , and two subsequences of length l in X , i.e. $S_{i,l}$ and $S_{j,l}$, the Euclidean Distance between them is defined as follows:

$$EuclideanDist(S_{i,l}, S_{j,l}) = \sqrt{\sum_{q=0}^{l-1} (x_{i-q} - x_{j-q})^2} \quad (1)$$

where $l \leq i, j \leq n$.

Next, we will give the definition of anomaly in time series. First we introduce the most similar non-self-match. Note that the anomaly here is not just for the abnormal points. It is a measure of the abnormal degree of every point in time series.

Definition 7. The most similar non-self-match. Given a subsequence $S_{i,l}$ of length l and a sliding window $W_{i,m}$ of length m , the most similar non-self-match $S'_{i,l}$ of $S_{i,l}$ is defined as the non-self-match with the smallest Euclidean distance of $S_{i,l}$ in $W_{i,m}$, that is:

$$S'_{i,l} = S_{j,l} \quad (2)$$

$$s.t. EuclideanDist(S_{i,l}, S_{j,l}) = \min\{EuclideanDist(S_{i,l}, S_{p,l})\}$$

where $S_{p,l}$ is the non-self-match of $S_{i,l}$ in $W_{i,m}$.

Definition 8. Anomaly. Given a time series $X = \{x_t, t = 1, 2, \dots, n\}$ with length n , and a subsequence $S_{i,l}$ which is expanded forward from x_i , the Anomaly of x_i is defined as the distance between $S_{i,l}$ and $S'_{i,l}$ in the sliding window $W_{i,m}$, that is: $Anomaly(x_i) = EuclideanDist(S_{i,l}, S'_{i,l})$, where $l \leq i \leq n$.

When the length of sliding window and subsequence are given, we can calculate the anomaly of each point in the time series, and the anomaly of the abnormal points in time series is larger than that of the normal points. When an abnormal situation occurs, the anomaly increases significantly. The definition of the abnormal point needs to be introduced.

Definition 9. Abnormal Point. Given a time series $X = \{x_t, t = 1, 2, \dots, n\}$ with length n , the length of subsequence l , the length of sliding window m , and the threshold K , x_i is an abnormal point iff $Anomaly(x_i) > K$.

2.2 Related Work on Anomaly Detection

Anomaly detection in time series can be applied in many fields and several algorithms have been proposed for various scenes including medicine [16], finance and the Internet. Some anomaly detection algorithms in time series use traditional analysis methods, including ARIMA [9], exponential smoothing [20], and time series decomposition [1, 19]. Other research results use statistical methods for anomaly detection, such as PCA [10], linear regression [6, 15], extreme value theory [17], median theory [5], etc. The rapid development of machine learning algorithms in recent years results in the usage of many machine learning algorithms such as neural network [8], SVDD [4], DBSCAN [15] and so on.

In industry, Internet companies develop time series analysis platforms based on their own business needs. Yahoo's time series analysis system EGADS [3] includes three modules: time series forecasting, time series anomaly detection and alerting module system, which integrates multiple time series analysis and anomaly detection algorithms. In 2014, Twitter proposed an anomaly detection algorithm based on time series decomposition and generalized Extreme Studentized Deviate test [1]. Twitter also put forward a complementary breakout detection algorithm in time series, and provided the R libraries [24, 26] for these two algorithms. Netflix developed the anomaly detection system Surus to ensure the validity of the company's data, and open sourced the algorithm RAD [21], which mainly uses RPCA [23] to detect abnormal points. LinkedIn's open source tool, luminol [22], is a lightweight library for time series analysis that supports anomaly detection and correlation analysis.

Both in academia and industry, the measurements of anomaly detection algorithms should include the following aspects:

- (1) Accuracy. Anomaly detection is essentially a two-category problem. For any point, a false classification may lead to some irreparable consequences. Therefore, the accuracy of the algorithm is the most important. When measuring the accuracy

of anomaly detection algorithms, the accuracy, recall, and the harmonic mean of the accuracy and recall, F1-Score are usually used.

- (2) Offline or online. This measurement should be considered in two aspects. First, in general, in order to maintain the accuracy of the anomaly detection algorithm, it is necessary to update the model and parameters continuously. Second, from the aspect of intelligent operation, the time series data obtained from the monitoring system is streaming, so a proper algorithm should detect the abnormal situation as the points flow in.
- (3) Super parameters. Most regression and classification algorithms need to set several super parameters, such as the K value in K-nearest-neighbor algorithm. The values of super parameters tend to have a decisive influence on the effect of the algorithm. The anomaly detection algorithm is no exception. If the number of super parameters is less, and the effect of the algorithm is less affected by the super parameter size, this algorithm is undoubtedly a more stable model.
- (4) Time complexity. The detection of anomalies needs to be both accurate and fast, so the time complexity of the algorithm also needs to be considered [13].
- (5) The main purpose of the anomaly detection algorithm is to be able to detect outliers in a time series accurately and quickly. Many existing algorithms model the normal points and find the anomaly by comparing the difference between the currently analyzed data and the normal data. If the difference is large enough, the current analyzed point is considered abnormal. Based on this idea, many data mining algorithms are applied to detect time series anomalies, such as linear regression [6, 15], support vector machine [4], neural network [8] and so on. These algorithms usually divide the obtained time series data into a training set and a test set, train the model in the training set and perform detection on the test set. In practical applications, time series data is streaming and changing constantly. Therefore, if the model obtained by training is not be updated in real time, it will be difficult to apply the model to new data. In addition, the variability of normal data and abnormal data makes it difficult to use a single model to detect anomalies accurately during the whole detection period. In summary, an online anomaly detection algorithm can not only run in streaming data, but also update the model itself continuously, which will be more suitable for the actual situation.

3 An Online Anomaly Detection Algorithm

This section introduces the distance-based online anomaly detection algorithm proposed in this paper. Firstly, the main idea of the algorithm is expounded, and then the proposed algorithm is given by formal definition.

3.1 Main Idea

By observing and analyzing the real time series anomaly data in operation, we find that abnormal points usually show a sudden increase or a sudden decrease compared with the surrounding points. The abnormal sequences usually show a different trend from its surrounding points (especially the previous points), as shown in Fig. 2. Figure 2 is a real operation database, which contains both abnormal sequences and abnormal points.

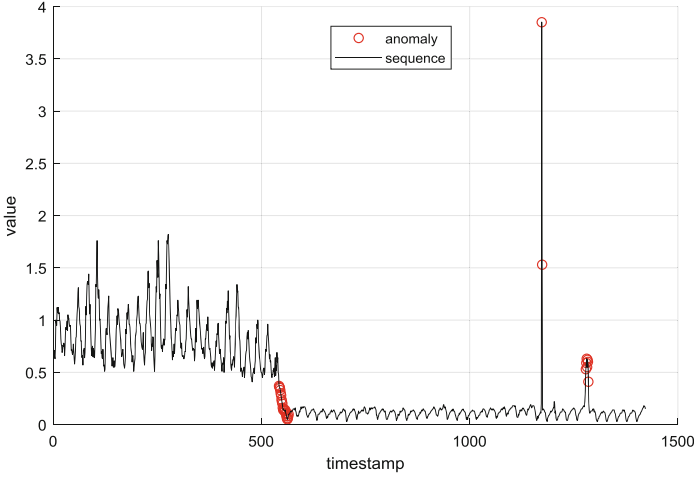


Fig. 2. A real operation data that contains both abnormal points and subsequences

According to the characteristics of abnormal data, we propose an online distance-based anomaly detection algorithm. The main idea is that for every point in the given time series, we calculate the anomaly of the subsequence expanded from that point in the sliding window. In general, the anomaly of abnormal data is significantly larger than that of normal data. Our algorithm uses two specific statistical distributions to fit the anomaly of each point in the time series. The anomaly of normal points often fluctuates around the distribution center, while the anomaly of abnormal points is so large that it is far from the center. Therefore, when the anomaly deviates from the center of the distribution too much, the point is regarded as an abnormal point.

3.2 The Anomaly in Sliding Window

Algorithm 1 *AnomalyCalculate* gives a calculation method for calculating the anomaly of each point in the time series by using a sliding window mechanism. The pseudo code is as follows.

Here we give the specific steps of *AnomalyCalculate*. Given a time series $X = \{x_t, t = 1, 2, \dots, n\}$ with length n , the length of sliding window m , the length of subsequence l , then for every point in the interval $[m, n]$, that is $\{x_i, i = m, m + 1, \dots, n\}$, we have the subsequence $S_{i,l}$, which is expanded from x_i and of length l , and the sliding window $W_{i,m}$, which is expanded from x_i and of length m , where $m \leq i \leq n$. We will find the non-self-match which has the least Euclidean distance with $S_{i,l}$ in $W_{i,m}$, i.e. the most similar non-self-match $S'_{i,l}$. The Euclidean distance between $S_{i,l}$ and $S'_{i,l}$, i.e. $EuclideanDist(S_{i,l}, S'_{i,l})$, is the anomaly of x_i , denoted as $Anomaly(x_i)$, just as the description in definition 8. In order not to compare the current data with the previous abnormal data, algorithm 1 restricts that the most similar non-self-match is normal data (Fig. 3).

After calculating the anomaly of each point in a given time series by Algorithm 1, it is found that when the abnormal point or the abnormal sequence appears, the anomaly

Algorithm 1 AnomalyCalculate

Input: Time Series (x_t) , Sliding Window length m , Subsequence length l

Output: $Anomaly(x_t)$

```

1: Function  $Anomaly(x_t) = AnomalyCalculate(x_t, m, l)$ 
2:  $Anomaly(x_t) = \vec{0}$ 
3: For  $i=m$  to  $n$ 
4:   nearest_neighbor_dist=infinity
5:   For  $j=i-m+1$  to  $i-l$ 
6:     IF  $S_{j,l}$  is not anomaly &  $EuclideanDist(S_{i,l}, S_{j,l}) < nearest\_neighbor\_dist$ 
7:       nearest_neighbor_dist= $EuclideanDist(S_{i,l}, S_{j,l})$ 
8:   End
9:  $Anomaly(x_t) \leftarrow nearest\_neighbor\_dist$ 
10:End

```

Fig. 3. AnomalyCalculate algorithm

will increase significantly. Figure 4 shows that the anomaly is non-negative, and after a period of m (the length of sliding window), the anomaly fluctuate randomly and increase significantly when the abnormal situation occurs.

For comparison, Fig. 4 also shows the anomaly calculated using the cosine distance and the Pearson correlation coefficient. Note that a proper distance metrics should maximum the anomaly of abnormal data and minimum the anomaly of normal data to separate these two categories. From Fig. 4 we can see Euclidean distance is the most appropriate method.

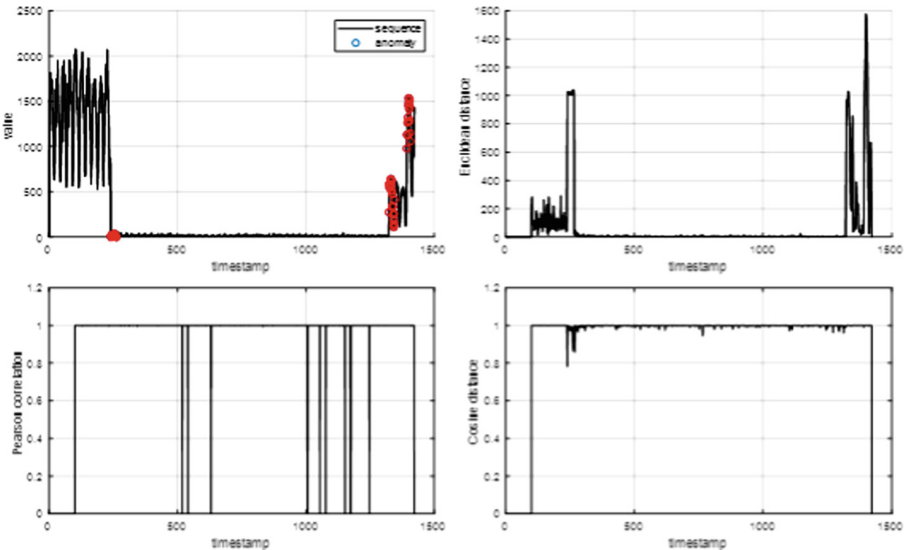


Fig. 4. The operation datasets and its corresponding anomaly line chart using Algorithm 1

3.3 Threshold Selection Mechanism

From the analysis in Sect. 3.2, AnomalyCalculate can maximum the anomaly of abnormal data. However, the operation data obtained in actual scene is streaming data and different time series have different orders of magnitude, so it is not enough to calculate the anomaly. The key is that we should select a threshold, and then if the anomaly of the current point exceeds the threshold, the point should be considered as abnormal. In this paper, we use distribution-fitting method to determine the threshold.

Assume that the anomaly of the points in a time series follows a certain distribution, then the anomaly of abnormal points are so large that they fall outside the $1 - \alpha$ ($0 < \alpha < 1$) quantile. Therefore, when the anomaly of the current point falls in that part, we consider it as abnormal. After observing the anomaly distribution of massive time series, we choose normal distribution and lognormal distribution to fit the anomaly.

In this paper, we use the recursive calculation method of mean and variance [14] to estimate the mean and variance of the distribution, as calculated below:

$$\begin{aligned} \mu_n &= \frac{1}{n}x_n + \frac{n-1}{n}\mu_{n-1} \\ S_n^2 &= \frac{n-1}{n}S_{n-1}^2 + \frac{n-1}{n^2}(x_n - \mu_{n-1})^2 \end{aligned} \quad (3)$$

For the addition of new elements, the forgetting factor λ is introduced into the recursive formula for calculating the mean and variance. Then the mean and variance with forgetting factor are calculated as follows:

$$\mu_n = \frac{\sum_{i=1}^n \lambda^{n-i} x_n}{\sum_{i=1}^n \lambda^{n-i}} \quad S_n^2 = \frac{\sum_{i=1}^n \lambda^{n-i} (x_i - \mu_n)^2}{\sum_{i=1}^n \lambda^{n-i}} \quad (4)$$

The recursive calculation methods of the mean and variance are as follows:

$$\begin{aligned} \mu_n &= \frac{1-\lambda}{1-\lambda^n} x_n + \frac{\lambda(1-\lambda^{n-1})}{1-\lambda^n} \mu_{n-1} \\ S_n^2 &= \frac{\lambda(1-\lambda^{n-1})}{1-\lambda^n} S_{n-1}^2 + \frac{\lambda(1-\lambda^{n-1})(1-\lambda)}{(1-\lambda^n)^2} (x_n - \mu_{n-1})^2 \end{aligned} \quad (5)$$

3.4 Anomaly Detect

AnomalyCalculate in Sect. 3.2 calculates the anomaly of the points in a given time series. By using the two fitting distributions given in Sect. 3.3 and the recursive calculation method of mean and variance, we can estimate and update the parameters of the two fitting distributions as we calculate the anomaly of each point. Because the mean and variance calculated in initial stages fluctuate greatly, which may lead to the instability of the fitting distribution and then affect the detection effect, the proposed algorithm AnomalyDetect introduces a transition period, in which the mean and variance are not updated. The initial values of mean and variance are calculated by the anomaly of points in the transition period, and the mean and variance after the transition period are updated by formula (3) or formula (5).

Algorithm 2 (AnomalyDetect) gives the specific steps of fitting the anomaly distribution and estimating the parameters while calculating the anomaly of each point in the given time series, delimiting the threshold and determining whether the current point is abnormal, as shown in Fig. 5.

Based on AnomalyCalculate, AnomalyDetect adds several steps including anomaly distribution fitting, parameter estimation, threshold delimiting, and anomaly detection. Figure 6 shows the specific implementation process of AnomalyDetect. As shown in Fig. 6, for the subsequence expanded forward from the current point, AnomalyDetect looks for the most similar non-self-match in the sliding window, which is also expanded forward from the current point. AnomalyDetect calculates the Euclidean distance of the two subsequences as the anomaly of the current point, and calculates the degree of deviation from the distribution center. When the deviation is excessive, the current point is considered an abnormal point. Similarly, in order to prevent comparing current subsequence with the abnormal points, we delimit the most similar non-self-match are all normal data.

Algorithm 2 AnomalyDetect

Input: Time Series (x_t) , Sliding Window length m , Subsequence length l , transition t , anomaly threshold $\alpha \in (0,1)$

Output: Anomaly flags \vec{f}

```

1: Function  $Anomaly(x_t) = AnomalyDetect(x_t, m, l)$ 
2:  $Anomaly(x_t) \leftarrow \bar{0}$ 
3: Anomaly flags  $\vec{f} \leftarrow \bar{0}$ 
4: transition_dist  $\leftarrow$  null array
5:
6: For  $i=m$  to  $n$ 
7:   nearest_neighbor_dist=infinity
8:   For  $j=i-m+l$  to  $i-l$ 
9:     IF  $S_{j:l}$  is not anomaly &  $EuclideanDist(S_{i:l}, S_{j:l}) < nearest\_neighbor\_dist$ 
10:    nearest_neighbor_dist= $EuclideanDist(S_{i:l}, S_{j:l})$ 
11:   End
12: End
13:  $Anomaly(x_t) \leftarrow nearest\_neighbor\_dist$ 
14: IF  $i < m+t$ 
15:   transition_dist+[nearest_neighbor_dist]
16: End
17: IF  $i = m+t$ 
18:    $\mu \leftarrow mean(transition\_dist)$ 
19:    $\sigma^2 \leftarrow var(transition\_dist)$ 
20: End
21: IF  $i \geq m+t$ 
22:   Calculate the  $\alpha$ -quantile  $z_\alpha$  of  $N(\mu, \sigma^2)$  or  $LN(\mu, \sigma^2)$ 
23:   IF nearest_neighbor_dist  $> z_\alpha$ 
24:     Anomaly flags  $\vec{f}_i \leftarrow 1$ 
25:   Else
26:     Update  $\mu$  and  $\sigma^2$ 
27:   End
28: End

```

Fig. 5. AnomalyDetect algorithm

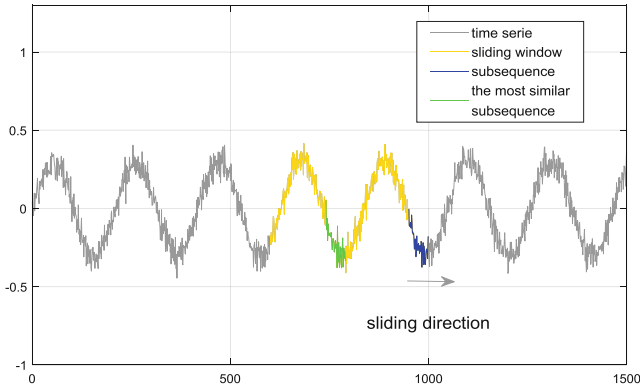


Fig. 6. The specific implementation of AnomalyDetect algorithm

3.5 Complexity Reduction

From the perspective of time complexity, AnomalyDetect calculates Euclidean distance for m times to get the most similar non-self-match. In some scenarios where data points are dense and high efficiency is required, we propose a mechanism to reduce the time complexity for our algorithm.

We use k-means and time-space trade-off mechanism. In the initial sliding window, k-means is used to cluster normal subsequences to get k classes and k class centers. The algorithm maintains two data structures in the detection process: an array is used to mark the classes of each subsequence in the sliding window, and a dictionary is used to store the class centers, as shown in Fig. 7.

| value0 | value1 | value2 | label |
|------------|------------|------------|-------|
| 52.6063 | 53.2957 | 186.8987 | 4 |
| 53.2957 | 186.8987 | 249.9720 | 4 |
| 186.8987 | 249.9720 | 279.8083 | 9 |
| 249.9720 | 279.8083 | 264.1173 | 9 |
| 279.8083 | 264.1173 | 352.9358 | 9 |
| ... | ... | ... | ... |
| -1253.8910 | -1287.2320 | -1229.9050 | 3 |
| -1287.2320 | -1229.9050 | -1291.1907 | 3 |
| -1229.9050 | -1291.1907 | -1267.7189 | 3 |
| -1291.1907 | -1267.7189 | -1281.5156 | 3 |
| -1267.7189 | -1281.5156 | -1347.3994 | 3 |

| | | | |
|-----|------------|------------|------------|
| 0 | 333.4046 | 327.2986 | 316.6243 |
| 1 | -1212.5169 | -1231.5428 | -1255.9541 |
| ... | ... | ... | ... |
| 9 | 80.9580 | 53.0702 | 45.2936 |

Fig. 7. Two data structures used to reduce time complexity

While the sliding window moves forward, we compare the current subsequence with class centers and determine it abnormal or not. If it is determined normal, the subsequence in the sliding window will be updated: the first subsequence in the sliding window is discarded, the current subsequence is added at the end, and the class centers of the class to which the two subsequences belong is updated.

4 Experiment Result and Analysis

In this section, several experiments of anomaly detection algorithm are carried out on the actual operation time series datasets, and the results are analyzed.

4.1 Data Preparation

This article uses the actual operation datasets [2] provided by Yahoo!, which contains four subfolders: A1, A2, A3, and A4. The A1 dataset contains 67 real operation time series, and the A2, A3, and A4 datasets each contain 100 pieces of synthetic time series. The synthetic time series is composed of periodicity, trend and noise, and the complexity of the synthetic time series is gradually increased. Figure 8 is the representative timing charts of the four datasets, where the abnormal points are marked in red.

The research on the synthetic data in Yahoo! S5 data shows that the A2, A3, and A4 datasets randomly change the normal points value to generate abnormal data points. The abnormal points are divided into two categories: anomaly and changepoint. The normal data in the A2 dataset is composed of simple trend, single periodic data and

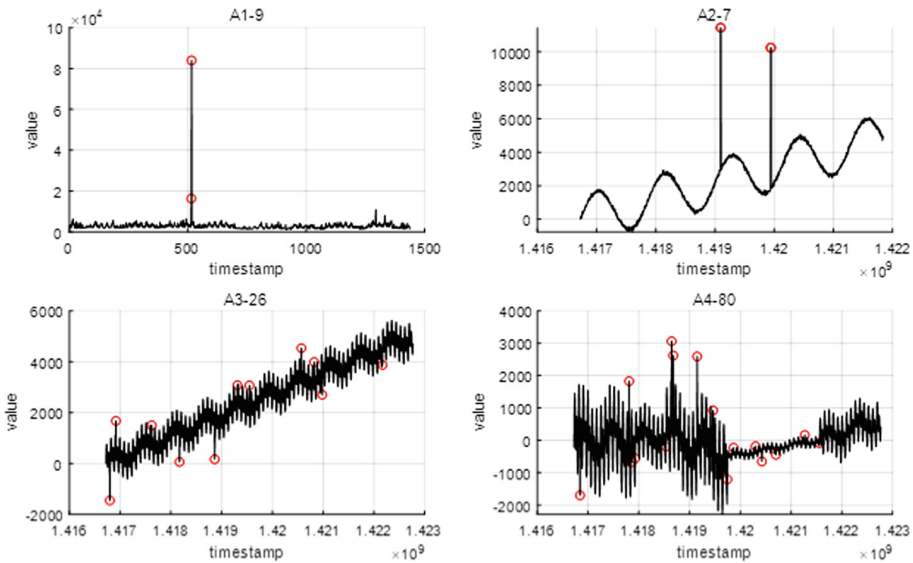


Fig. 8. Yahoo Webscope S5 Datasets (Color figure online)

noise. The abnormal point type is only anomaly. The normal data in the A3 data set consists of a single trend, three periodic data with different cycles and amplitudes, and noise. The abnormal point type is only anomaly. The A4 data set is formed of a trend and three periodic data with different cycles and varying amplitudes, and noise. The abnormal point type includes both anomaly and changepoint. Figure 9 shows the synthetic time series in the A3 and A4 datasets and the trend and period data, labeled with different colors.

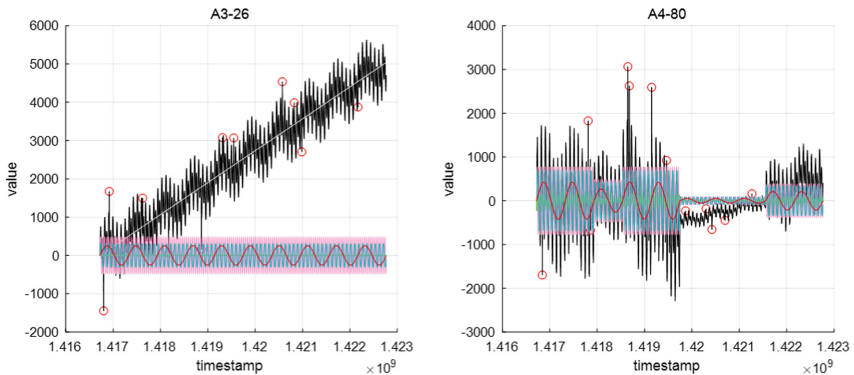


Fig. 9. The synthesis sequence and decomposition sequence of A3 and A4 datasets

4.2 Data Preparation

Anomaly detection problem in time series can be regarded as a two-category problem. Each point in the time series is marked as normal or abnormal, and can also be predicted as normal or abnormal. If the normal is marked as 0, the abnormal flag is 1, then the real situation and the prediction situation can be counted as a contingency table, that is, the confusion matrix, as shown in Table 1.

Table 1. Confusion matrix

| Actual | Predicted | |
|--------------|---------------------|---------------------|
| | Positive(1) | Negative(0) |
| Positive (1) | True Positive (TP) | False Negative (FN) |
| Negative (0) | False Positive (FP) | True Negative (TN) |

Each point in the time series can be labeled as TP, FN, FP, and TN.

F1-Score is another indicator to measure the accuracy of the two-category model. It is the harmonic mean of the precision and recall and considers both of these two evaluation methods.

4.3 Data Preparation

In this paper, we use AnomalyDetect to perform experiments on the Yahoo! S5 datasets. The detection of abnormal data will be performed after a sliding window plus a transition period. While detecting the abnormal points, with the fluctuation of the mean and variance, the anomaly distribution will have a few changes. At the end of the detection process, we can get the anomaly of each point in the given time series, and the anomaly distribution estimated by the final iteration.

Table 2 lists the detection effects of AnomalyDetect in Yahoo! S5 datasets, where each time series data is fixed with a sliding window size (200), a time subsequence length (3), a transition period (50), and a forgetting factor (1). The actual detection process is after a sliding window and a transition period. Table 4 gives the precision, recall and F1-Score of the four datasets.

Table 2. The detection effect of Algorithm 2 on Yahoo! Webscope S5 dataset

| Dataset | Algorithm | TP | FP | FN | Precision | Recall | F1 Score |
|---------|-----------|------|----|-----|-----------|--------|----------|
| A1 | norm | 1543 | 10 | 111 | 0.9329 | 0.9936 | 0.9623 |
| | lognorm | 1543 | 10 | 111 | 0.9329 | 0.9936 | 0.9623 |
| A2 | norm | 461 | 0 | 5 | 0.9893 | 1 | 0.9946 |
| | lognorm | 461 | 1 | 5 | 0.9893 | 0.9978 | 0.9935 |
| A3 | norm | 766 | 30 | 37 | 0.9539 | 0.9623 | 0.9581 |
| | lognorm | 766 | 29 | 37 | 0.9539 | 0.9635 | 0.9587 |
| A4 | norm | 572 | 66 | 315 | 0.6449 | 0.8966 | 0.7502 |
| | lognorm | 577 | 68 | 281 | 0.6725 | 0.8946 | 0.7678 |

As can be seen from Table 2, the detection accuracy of AnomalyDetect is very high, reaching more than 0.95 on the A1–A3 dataset, especially in the real operation dataset A1, the F1-Score also reached 0.96. And in A4 dataset, it reaches more than 0.75. This is the case with the basic ideas of AnomalyDetect. AnomalyDetect is mainly based on the comparison of morphological similarity of different subsequences in time series, not just the correlation of adjacent points. Moreover, AnomalyDetect can realize the universal detection of both abnormal points and abnormal sequences by adjusting the length of subsequences.

Similarly, AnomalyDetect has its shortcomings. Its detection effect on the A4 dataset is not perfect, which is related to the complex and varied form of the A4 dataset.

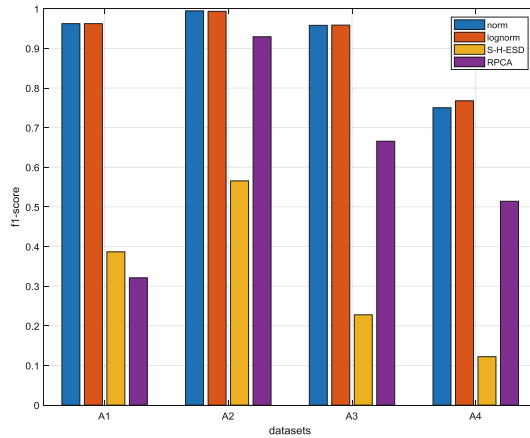
In this article, we use two open source detection algorithms for comparison. Table 3 lists the detection accuracy of the Twitter anomaly detection algorithm S-H-ESD [24] and Netflix’s anomaly detection algorithm RPCA [21] on the Yahoo! Webscope S5 datasets.

The S-H-ESD algorithm is an offline algorithm based on time series decomposition and generalized ESD test. RPCA is also an offline algorithm, mainly based on the Robust PCA algorithm. Both algorithms need to obtain the whole time series data, and

Table 3. The Results of S-H-ESD and RPCA on Yahoo Webscope S5 datasets

| Dataset | Algorithm | TP | FP | FN | Precision | Recall | F1-score |
|---------|-----------|-----|------|------|-----------|--------|----------|
| A1 | S-H-ESD | 571 | 1147 | 664 | 0.4623 | 0.3324 | 0.3867 |
| | RPCA | 636 | 1033 | 1655 | 0.2776 | 0.3811 | 0.3212 |
| A2 | S-H-ESD | 140 | 172 | 43 | 0.765 | 0.4487 | 0.5657 |
| | RPCA | 447 | 19 | 49 | 0.9012 | 0.9592 | 0.9293 |
| A3 | S-H-ESD | 86 | 583 | 0 | 1 | 0.1286 | 0.2278 |
| | RPCA | 509 | 434 | 77 | 0.8686 | 0.5398 | 0.6658 |
| A4 | S-H-ESD | 112 | 454 | 1156 | 0.0883 | 0.1979 | 0.1221 |
| | RPCA | 432 | 405 | 411 | 0.5125 | 0.5161 | 0.5143 |

then the abnormal points are detected. Drawing the F1-score of the four algorithms on the Yahoo S5 dataset into a bar graph can compare the detection effects of the four algorithms more clearly, as shown in Fig. 10.

**Fig. 10.** The F1-score of different algorithms on Yahoo Webscope S5 datasets

As can be seen from Fig. 11, AnomalyDetect is significantly better than another two algorithms in term of accuracy. From the perspective of the datasets, the four algorithms perform best in the A2 dataset because of the obvious periodicity of the A2 dataset. If the trend is more obvious, the noise impact is smaller, and the abnormal points are more different from the normal points, then the detection is easier. In addition, as can be seen from Fig. 11, AnomalyDetect has a significant advantage in the A1 dataset. A1 is a real operation dataset. Unlike the synthetic dataset, the timing pattern of A1 is more complex and reflects a real operation situation. And the innovative distance-based algorithm AnomalyDetect is more suitable for real situation data.

In summary, compared with other anomaly detection algorithms, the algorithm shows obvious advantages in the following aspects:

- (1) Accuracy. It can be seen from Tables 2 and 3 that the F1-score of the algorithm on the four data sets is relatively high.
- (2) Versatility. Usually a single anomaly detection algorithm is not applicable to all time series data. For example, the algorithm for detecting abnormal points does not detect abnormal sequences well. However, based on the similarity between different subsequences, AnomalyDetect can detect both abnormal points and abnormal sequences. What's more, the detection effect of AnomalyDetect on real time series is not inferior to that of synthetic time series.
- (3) Online. The analysis in Sect. 2.2 indicates that the online characteristic of a time series anomaly detection algorithm is an inevitable requirement for ensuring the real-time and accuracy. Therefore, AnomalyDetect's online update mechanism makes it significantly better than other algorithms.

Similarly, AnomalyDetect also has its inferior performance. The algorithm involves about four super parameters and is sensitive to the super parameter size. Because the algorithm needs to delimit the threshold, and the model parameters need to be updated continuously, the size of the super parameter plays an important role. As analyzed in Sect. 2.2, a better and more robust algorithm should have less dependence on the super parameters setting, but AnomalyDetect is slightly inferior in this respect.

5 Summary and Prospect

In this paper, a new online anomaly detection algorithm is proposed for the real scene. The algorithm judges whether the point or sequence is abnormal by calculating the anomaly of the subsequence in the time series. It has the characteristics of linear and real-time updating, etc., in Yahoo. Experiments on the Webscope S5 dataset show that the algorithm has high accuracy.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant No. 61672384), Fundamental Research Funds for the Central Universities under Grants No. 0800219373.

References

1. Vallis, O., Hochenbaum, J., Kejariwal, A.: A novel technique for long-term anomaly detection in the cloud. In: HotCloud (2014)
2. Yahoo: S5 - A Labeled Anomaly Detection Dataset, version 1.0 (2015). <http://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70>
3. Laptsev, N., Amizadeh, S., Flint, I.: Generic and scalable framework for automated time-series anomaly detection. In: KDD, pp. 1939–1947 (2015)
4. Huang, C., Min, G., Wu, Y., Ying, Y., Pei, K., Xiang, Z.: Time series anomaly detection for trustworthy services in cloud computing systems
5. Sagoolmuang, A., Sinapiromsaran, K.: Median-difference window subseries score for contextual anomaly on time series. In: IC-ICTE (2017)

6. Thill, M., Konen, W., Bäck, T.: Online anomaly detection on the webscope S5 dataset: a comparative study. In: EAIS (2017)
7. Chen, Y., Hu, B., Keogh, E., Batista, G.E.: DTW-D: time series semi-supervised learning from a single example
8. Suh, S., Chae, D.H., Kang, H.-G., Choi, S.: Echo-state conditional variational autoencoder for anomaly detection. In: International Joint Conference on Neural Networks (IJCNN) (2016)
9. Yu, Q., Jibin, L., Jiang, L.: An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **12**(1), 9653230 (2016)
10. Hyndman, R.J., Wang, E., Laptev, N.: Large-scale unusual time series detection
11. Wei, L., Kumar, N., Lolla, V., Keogh, E., Lonardi, S., Ann Ratanamahatana, C.: Assumption-free anomaly detection in time series. In: 17th International Conference on Scientific and Statistical (2005)
12. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. AAAI Technical Report WS-94-03
13. Chandola, V., Cheboli, D., Kumar, V.: Detecting anomalies in a time series database. Department, University of Minnesota, Technical report 12 (2009)
14. Welford, B.P.: Note on a method for calculating corrected sums of squares and products. *Technometrics* **4**(3), 419–420 (1962)
15. Watson, S.M., Tight, M., Clark, S., Redfern, E.: Detection of outliers in time series. Institute of Transport Studies, University of Leeds
16. Keogh, Eamonn, Lin, Jessica, Lee, Sang-Hee, Van Herle, Helga: Finding the most unusual time series subsequence: algorithms and applications. *Knowl. Inf. Syst.* **11**(1), 1–27 (2007)
17. Siffer, A., Fouque, P.A., Termier, A., Largouët, C.: Anomaly detection in streams with extreme value theory. In: Proceedings of the 23rd ACM SIGKDD International Conference (2017)
18. Lin, J., Keogh, E., Lonardi, S. Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (2003)
19. Chen, Y., Mahajan, R., Sridharan, B., Zhang, Z.-L.: A provider-side view of web search response time. In: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, vol. 43 (2013)
20. Yan, H.: Argus: end-to-end service anomaly detection and localization from an ISP's point of view. In: IEEE INFOCOM (2012)
21. Netflix: Surus. <https://github.com/Netflix/Surus>
22. LinkedIn: luminol. <https://github.com/linkedin/luminol>
23. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis (2011)
24. Twitter: Anomaly Detection. <https://github.com/twitter/AnomalyDetection>
25. Laptev, N., Amizadeh, S.: Yahoo anomaly detection dataset S5 (2015). <http://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70>
26. Twitter: Breakout Detection. <https://github.com/twitter/BreakoutDetection>
27. Zhang, S., et al.: Rapid and robust impact assessment of software changes in large internet-based services. In: CoNEXT 2015, 01–04 December 2015, Heidelberg, Germany (2015)