Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3187697, 12 pages
https://doi.org/10.1155/2020/3187697

Research Article

Hydrologic Time Series Anomaly Detection Based on Flink

Feng Ye,[1] **Zihao Liu**,[2] **Qinghua Liu**,[2] **and Zhijian Wang**[1]

[1]*School of Computer and Information, Hohai University, Nanjing, China*
[2]*School of Computer, Jiangsu University of Science and Technology, Zhenjiang, China*

Correspondence should be addressed to Qinghua Liu; liuqh@just.edu.cn

Received 22 November 2019; Revised 9 April 2020; Accepted 11 April 2020; Published 28 May 2020

Academic Editor: Jean Jacques Loiseau

The data mining and calculation of time series in critical application is still worth studying. Currently, in the field of hydrological time series, most of the detection of outliers focus on improving the specificity. To efficiently detect outliers in massive hydrologic sensor data, an anomaly detection method for hydrological time series based on Flink is proposed. Firstly, the sliding window and the ARIMA model are used to forecast data stream. Then, the confidence interval is calculated for the prediction result, and the results outside the interval range are judged as alternative anomaly data. Finally, based on the historical batch data, the *K*-Means++ algorithm is used to cluster the batch data. The state transition probability is calculated, and the anomaly data are evaluated in quality. Taking the hydrological sensor data obtained from the Chu River as experimental data, experiments on the detection time and outlier detection performance are carried out, respectively. The results show that when calculating the tens of millions of data, the time costed by two slaves is less than that by one slave, and the maximum reduction is 17.43%. The sensitivity of the evaluation is increased from 72.91% to 92.98%. In terms of delay, the average delay of different slaves is roughly the same, which is maintained within 20 ms. It shows that, under big data platform, the proposed algorithm can effectively improve the computational efficiency of hydrologic time series detection for tens of millions of data and has a significant improvement in sensitivity.

1. Introduction

Hydrological data are divided into various types of hydrological time series according to their physical quantities. At present, many experts believe that hydrological time series is generally composed of determined and random components. The definite component has certain physical concept, and the random component is produced by the irregular oscillation and the stochastic influence [1]. Hydrological time series mainly shows the complex characteristics of randomness, fuzziness, nonlinearity, nonstationary, and multitime scale change [2].

In practice, as the world gets more instrumented and connected, we are witnessing a flood of digital data generated from diversified hardware (e.g., sensors) or software in the format of big data. With the development of informatization, the hydrologic stations accumulate a great deal of important data which contains many outliers. For hydrological time series, it can be judged as an anomaly with a large difference from the general law [3]. Outliers often contain important information, and it is greatly significant for subsequent analysis decisions by accurately finding the hidden value behind the data. At present, for hydrological time series, traditional methods are only applicable to small datasets, not to the current big data environment. Moreover, the accuracy only reaches the level of 99% in specificity [4], and the sensitivity still has room for improvement. With the increase in the amount of data, how to calculate efficiently has become a problem that cannot be ignored. The anomaly detection and calculation of time series in critical application is still worth studying. This paper presents an anomaly detection method for hydrological time series based on Flink. Firstly, the sliding window and the ARIMA model are used to forecast data stream on the Flink platform. Then, the confidence interval is calculated for the prediction result, and the results outside the interval range are judged as temporary anomaly data. Finally, based on the historical batch data, the *K*-Means++ algorithm is used to cluster the batch data, the state transition probability is calculated, and the anomaly data are evaluated in quality. This method can effectively improve the computational efficiency and gives a reliable confidence degree to enhance the overall sensitivity. The outliers are detected quickly and accurately in massive hydrological time series.

The following contents are organized as follows: Section 2 discusses the research work related to this paper; Section 3 introduces the methodology of hydrologic time series anomaly detection in detail; in Section 4, we continue to use the real hydrological sensor data as experimental data to verify the effectiveness of the proposed method. Finally, the summary and prospect are given.

2. Related Work

2.1. Anomaly Detection. Outliers [5] are data that deviate most of the data in the dataset, which is not suspected of being a random error, but arises from a completely different mechanism. The following are some of the main methods of anomaly detection.

Niu [6] proposes a short-term electricity price hybrid forecasting model based on wavelet transform and ARIMA, which can detect the mutation point. The model can indeed detect the situation of abrupt change point, but it is insufficient for the nonlinear part or the data with too long time series. Gil [7] presents an anomaly detection based on vector machine and principal element analysis. Firstly, the principal element analysis method is used to reduce the latitude, and then, the SVM is used to model and examine the anomaly data. But if there are more kinds of outliers, the detection accuracy is not good. Sun [3] proposes a hydrological time series anomaly value detection based on ARIMA-SVR, which uses ARIMA to predict the linear parts, and SVM predicts the nonlinear part. It adds the predicted result and evaluates the value not in the confidence interval as the anomaly value. This kind of algorithm has a good effect on small-scale datasets, but it is not able to deal with massive data or multivariate data. It is also difficult to determine the threshold value.

The method based on distance detection is to set some distance function to calculate the distance of the data point. When the distance between one point and the other point is too large, it is regarded as the anomaly point. Vy and Anh [8] present an anomaly detection algorithm with variable length in time series. Firstly, it segments a time series and then calculates the anomaly factors of each mode. After that, the distance between them is calculated. At last, the anomaly is judged by the distance of the abnormal factors. The advantage of this method is that it is easy to use; the time complexity is relatively small, but it is not sensitive to local anomaly points.

Ali et al. [9] propose the concept of local outlier factor (LOF) to compute the dataset density. The possibility that the object is an outlier is positively correlated with the LOF. However, the mix of different densities will result in the detection error. Although some related improvement schemes are proposed, the overall time complexity is higher.

The clustering algorithm [10] divides the points in the time series into several clusters, and these points which do not belong to any cluster will be regarded as the anomaly, but the time series has a trend characteristic and cannot be classified as cluster analysis simply. Therefore, the clustering algorithm is too dependent on cluster quality, which leads to low accuracy and efficiency.

Hypothesis testing is a method for discovering anomaly samples. The dataset obeys a known distribution or probability model. If a point in the dataset is inconsistent with its distribution, the anomaly is judged. Twitter opens source the traffic anomaly detection algorithm S-H-ESD in 2015 [11], and the algorithm is to use STL to decompose the sequence and investigate the residuals. Assuming this is in accordance with the normal distribution, the outlier can be extracted using generalized ESD. However, if the feature distribution is unknown, the priori hypothesis does not necessarily effect. Then, the error detection rate of this method is high, and it cannot adapt to the multivariate time series well.

Yu [4] proposes the hydrological time series anomaly detection based on sliding window prediction, but the computational complexity is high. Yang et al. [12] use the knowledge granularity method to find the abnormal data in time series, reduce the time cost of the detection process, and improve the detection efficiency. Liu and Wang [13] propose an anomaly factor detection method based on the extremum difference, the slope, and the mean value. Good results are obtained.

In recent years, the accuracy of model prediction is also the goal of many researchers, and Zeng et al. [14] propose fuzzy forecasting based on linear combinations of independent variables, subtractive clustering algorithm, and artificial bee colony algorithm to predict time series. The proposed method gets higher forecasting accuracy rates than the existing methods for forecasting the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX), the enrolment of the University of Alabama, and the daily percentage of $CO_{2.}$ In addition, Zeng et al. [15] still propose interval-valued intuitionistic fuzzy multiple attribute decision-making based on nonlinear programming methodology and TOPSIS method. The method can overcome the drawbacks of the MADM methods. It offers us a very useful way to deal with MADM problems in IVIF environments. Although these two methods have extremely high accuracy, the computational efficiency in the big data is still a problem worth studying.

2.2. Flink. Apache Flink is a framework and distributed processing engine for stateful computation of unbounded and bounded data flows. Flink is designed to run in all common clustered environments, performing calculations at memory speed and any size. The Flink centralized swarm mode deployment on yarn is undergone through the construction of four centralized swarms in Figure 1.

Internally, Apache Flink represents job definitions using directed acyclic graphs (DAGs) [16]. The nodes of the graph are either sources, sinks, or operators. Source nodes read in or generate the input data, while sink nodes produce the

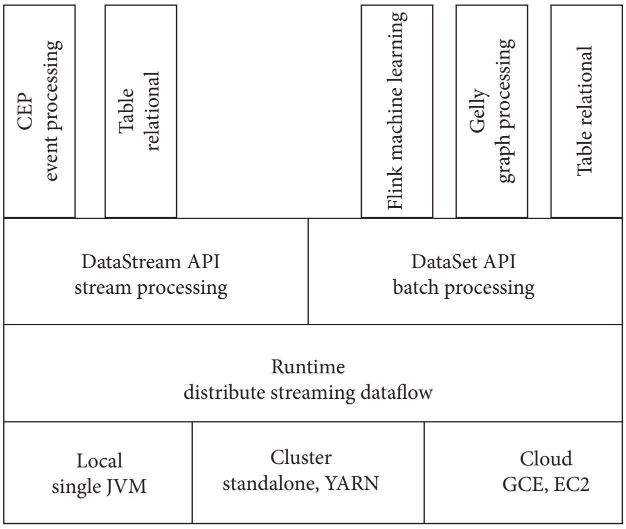

Figure 1: Flink architecture.

output. The inner vertices are operators which execute arbitrary user-defined functions (UDFs) that consume input from incident nodes and provide input for adjacent nodes. The DAGs generated from the user's job definitions are then transformed into the more concrete execution graphs, which contain the necessary information for running the job on a cluster. Data partitioning enables the data-parallel execution of the subtasks. During the transformation, the UDFs are split up into multiple parallel subtasks. Each subtask executes the same UDF. However, each process processes a different part of the input data. This setup makes clear where overheads might emerge. Firstly, the task scheduling as well as the deployment of the tasks on the respective machines introduce overhead. Moreover, the nodes must communicate with each other in order to distribute the workload. The communication between different nodes demand for serialization and transport buffering, which adds to the overhead.

On top of the possibility to define a job by using a DAG, a layer of second-order functions is implemented in order to simplify the development for the user [17]. Apache Flink provides APIs for implementing batch as well as stream processing. The exposed functions for dataset and data stream transformations are like functions known from functional programming (e.g., map, reduce, and filter). Additionally, the DataSet API provides transformations known from relational databases like joins and grouping. The DataStream API provides additional operators which are useful in the streaming context. These operators include the definition of windows and window-based aggregations.

3. The Proposed Methodology

3.1. Anomaly Detection Based on Sliding Window

3.1.1. The Definition of Slide Window. To define the sliding neighbor window L_i of the hydrological time series X to be detected point X_i. To reduce the complexity of the algorithm, the first L point of the point is used as the input parameter of the predictive model. This algorithm selects the left neighbor window of the prediction node as the algorithm input, and the unilateral definition is as follows:

$$L_{xi} = \left\{x_{i-L}, x_{i-L+1}, \ldots, x_{i-1}\right\}. \qquad (1)$$

3.1.2. Anomaly Detection. The core is to establish the ARIMA model [16] and predict the value of the observation point by sliding window input and then get a series of predicted values. ARIMA is a time series prediction model with autoregressive (AR) and moving average (MA). It is very flexible. The model becomes MA (q) when AR equals 0, while when MA $= 0$, ARIMA becomes AR (p) and specifically is described as

$$y(t) = v + \varphi_1 y(t-1) + \cdots + \varphi_p y(t-p) + \varepsilon(t), \quad (n \leq m). \qquad (2)$$

In (2), $\varepsilon(t)$ is the error. If $\varepsilon(t)$ is autocorrelation, then MA (q) can be represented as

$$\varepsilon(t) = a(t) + \Theta_1 a(t-1) + \cdots + \Theta_1 a(t-p). \qquad (3)$$

In (3), $\Theta_J \ (j = 1, 2, \ldots, q)$ is the parameters estimated; $a(t)$ is the white noise.

According to the above, ARIMA (p, q) is

$$\begin{aligned} y(t) = {} & v + \varphi_1 y(t-1) + \cdots + \varphi_p y(t-p) + \varepsilon(t) + v + \varphi_1 y(t-1) \\ & + \cdots + \varphi_p y(t-p) + a(t) + \Theta_1 a(t-1) + \cdots + \Theta_p a(t-q). \end{aligned} \qquad (4)$$

If the order n is relatively large, then AR (n) can be approximately equivalent to ARIMA (p, q); then,

$$y(t) = v + \sum_{i=1}^{n} \varphi_i y(t-i) + a_n(t). \qquad (5)$$

In equation (5), $a_n(t)$ is the error term of order n. The estimate of $a_n(t)$ can be achieved as follows:

$$\widehat{a}_n(t) = y_t + \sum_{i=1}^{n} \widehat{\varphi}_i y(t-i) - v. \qquad (6)$$

In equation (6), $\widehat{\varphi}_i$ can be achieved by least square estimation. Making use of $\widehat{a}_n(t)$, ARIMA (p, q) can be established:

$$y(t) = \left[v, \varphi_1, \ldots, \varphi_q \colon \Theta_1, \ldots, \Theta_q\right] \begin{bmatrix} 1 \\ y(t-1) \\ \cdots \\ \widehat{a}_n(t-1) \\ \cdots \\ \widehat{a}_n(t-q) \end{bmatrix} + a(t). \qquad (7)$$

ARIMA parameters n, p, and q are determined by AIC criterion:

$$\mathrm{AIC} = \ln\left|\sum_{a}(p,q)\right| + \frac{2}{S}P_{\mathrm{num}}. \tag{8}$$

In (8), S is the number of samples, and $\sum_a(p,q)$ is the determinant of Σ_a, which is the covariance matrix of $\widehat{a}(t)$.

First, unit root test is conducted for the time series . If it is a nonstationary sequence, it needs to be transformed into a stationary sequence by difference. Based on the AIC criterion, we need to determine the autoregressive order p and the moving average order q and find the p and q combination with the minimum AIC value. The ARIMA model is for nonstationary time series. It is applicable to hydrological time series. This paper takes the confidence interval of 95%. The confidence interval calculated by the ARIMA model is compared with the original sequence, and the value not in the confidence interval is judged as outliers.

3.2. Anomaly Value Verification.

After the exception values are identified by sliding window and ARIMA model, we also need to identify the confidence of the anomaly to determine whether the point is indeed an anomaly and reduce the error and the amount of work.

3.2.1. K-Mean++ Model.

K-Mean is one of the clustering algorithms. The principle of the *K*-Means++ algorithm is given the value of *K*, *K* represents the number of categories to divide the data into and then according to the similarity between the data to divide the data into *K* classes. The method of measuring the similarity of data is usually measured by the distance between data points, such as European distance, Hamming distance, and Manhattan distance.

It is a common practice to use the Euclidean distance to measure the similarity between the data. For example, for two points on the two-dimensional plane $A(x_1, y_1)$ and $B(x_2, y_2)$, the Euclidean distance between the two is

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{9}$$

Generally, a cluster is described by the centre of all points in the cluster, which is also called centroid. The way of computing the centre of mass is to calculate the mean of all data points in a cluster. The advantage of *K*-Means++ lies in the choice of the centre of mass. The *K*-Means++ model is optimized for the centroid uncertainty in the *K*-Means algorithm and chose the centroid K in the following strategy: hypothesis has selected the initial clustering centre n $(0 < n < K)$. When the clustering centre is in the selection of $n + 1$, the farther the distance from the current n clustering centre point will be a higher probability of being selected as the clustering centre $n + 1$. However, the selection of the first cluster centre $(n = 1)$ also adopts the random method.

Specifically, *K*-Means takes the entire time series $\{x_1, x_2, \ldots\}$ as input and sequence $T = \{T_1, T_2, \ldots\}$ as an output; the points on the time series are converted to various cluster points, and $\{T_1, T_2, \ldots\}$ indicates which centre this time point is in, which classify the time series in different states.

After providing the *K*-means++ model with the previous outliers and the value of their previous moment as input, the distance between each sample and the centre of the cluster is computed, and then the cluster centre is assigned to each sample.

3.2.2. State Transition Probability Matrix.

Markov processes in which time and state are discrete are called Markov chains, and it is assumed that

$$X_n = X(n), \quad n = 1, 2, \ldots. \tag{10}$$

It can be viewed as the result of successive observations of discrete Markov processes on the time set:

$$T_1 = \{0, 1, 2, \ldots\}. \tag{11}$$

The state space of the chain can be described that

$$I_1 = \{a_1, a_2, a_3, \ldots\}. \tag{12}$$

In the state of chain, Markov property is usually expressed by conditional distribution law. That is, satisfy

$$\forall n \in Z, r \in Z, 0 \leq t_1 < t_2 < \cdots < t_r < m; t_j, \quad m, n + m \in T_1. \tag{13}$$

$$\begin{aligned} &P\left\{X_{m+n} = a_j \,\middle|\, X_{t1} = a_{i1}, X_{t2} = a_{i2}, \ldots, X_{tr} = a_{ir}, X_m = a_i\right\} \\ &\quad = P\left\{X_{m+n} = a_j \,\middle|\, X_m = a_i\right\}, \quad a_i \in I. \end{aligned} \tag{14}$$

We sign equation (14) as follows:

$$P_{ij}(m, m + n), \tag{15}$$

and describe the conditional probability as

$$P\{m, m + n\} = P\left\{X_{m+n} = a_j \,\middle|\, X_m = a_i\right\}, \quad a_i \in I. \tag{16}$$

It means the transition probability of Markov chain from $m + n$ to a_j under the condition that m is in the state a_i at time m. Since the chain starts from any state a_i at time m to another state $m + n$, it must move to one of the states in $a_1, a_2, \ldots,$ so

$$\sum_{j=1}^{\infty} P_{ij}(m, m + n) = 1, \quad i = 1, 2, \ldots. \tag{17}$$

The matrix composed of transition probabilities is called the transition probability matrix of Markov chain:

$$P(m, m + n) = \left(P_{ij}(m, m + n)\right), \tag{18}$$

and the above formula (18) knows that the sum of the elements in each row of this matrix is equal to 1. When the transition probability is only related to $P_{ij}(m, m + n)$, i, j, and time interval n, it is denoted as P_{ij} or

$$P(m, m + n) = P_{ij}(n). \tag{19}$$

The transition probability is said to be stable. It is also homogeneous or time-homogeneous. In the case of homogeneous Markov chain, the transition probability is

$$P_{ij}(n) = P\left\{X_{m+n} = a_j \,\middle|\, X_m = a_i\right\}. \tag{20}$$

Equation (20) is called the n-step transition probability of Markov chains, and when $n = 1$, it is the one-step transition probability, which is particularly important. The matrix consisting of the one-step transition probability is called the one-step transition probability matrix:

$$P = \begin{bmatrix} p_{1,1} & p_{1,1} & \cdots & p_{1,1} & \cdots \\ p_{1,1} & p_{1,1} & \cdots & p_{1,1} & \cdots \\ M & M & O & M & O \\ p_{1,1} & p_{1,1} & \cdots & p_{1,1} & \cdots \\ M & M & O & M & O \end{bmatrix}, \tag{21}$$

$$p_{i,j} = p_r(j \,|\, i).$$

The following is the realization method of one-step transition probability matrix:

Step 1. Calculate the probability of each state:

$$P\{X_m = a_i\}. \tag{22}$$

Step 2. Calculate the probability of state a_i at moment m and state a_j at moment $m+1$:

$$P\left\{X_{m+n} = a_j, X_m = a_i\right\}. \tag{23}$$

Step 3. Compute transition probability:

$$\begin{aligned} P_{ij} &= P_{ij}(1) = P\left\{X_{m+1} = a_j \,\middle|\, Xm = a_i\right\} \\ &= \frac{P\left\{X_{m+n} = a_j, X_m = a_i\right\}}{P\{X_m = a\}_i}. \end{aligned} \tag{24}$$

Means model training is completed, the state sequence $\{T_1, T_2, \ldots\}$ can be obtained, and a state transition matrix can be obtained by calculation. To facilitate the calculation, it is necessary to convert the matrix to a data frame. There are three columns: the first column represents the state i, the second column represents the state j, and the third column represents the probability that the state i transfers to state j. Suppose a time series $\{x_i, x_{i+1}\}$ is transformed by K-Means++ model and it becomes the state sequence $\{T_i, T_j\}$, $x_i + 1$ appears after x_i, which in other words is equivalent to a transfer from state T_i to T_j, and the transfer probability is

$$p_{ij} = \frac{\text{the number of } T_i \text{ to } T_j}{\text{the number of } T_i \text{ to other state}}. \tag{25}$$

3.2.3. Outliers Check. In this paper, the probability of a transition is compared with the probability of the state of the previous moment being transferred to the next most probable state as an evaluation criterion. If x_i is to be inspected anomaly, the value of the previous moment is x_{i-1}, and the state of T_i and T_{i-1}, T_{i-1} is the most likely to be transferred to T_m; thus, we define an anomaly value probability:

$$p_i = 1 - \frac{\text{the possibility that } T_{i-1} \text{ to } T_i}{\text{the possibility that } T_{i-1} \text{ to } T_m}. \tag{26}$$

From the above, it is known that the probability of state T_i to T_m is a constant, the smaller the probability of state T_{i-1} transferring to T_i, the greater the p_i, the greater the probability of x_i being abnormal value.

In the reality of hydrological data anomaly judgement standard, the data more than 2 cm are judged as outliers. Using the clustering algorithm may appear abnormal values and the moment before its values in the same condition and lead to false negatives. So after the judgement, outliers with a probability of 0 but a difference of more than 2 cm should also be judged as outliers.

3.3. The Anomaly Detection Method. The hydrological time series anomaly detection algorithm based on Flink is combined with two processes: the anomaly detection and result verification. First, the ARIMA model of time series $\{x_1, x_2, \ldots, x_n\}$ is established by the idea of predictive detection by sliding window, and the predicted confidence interval is obtained, which is compared with the original data to identify the outliers. After the anomaly value is detected, the original data are clustered by K-Means++ algorithm, and the state transition matrix is computed after clustering. At last, the abnormal value is evaluated by the state transition and finally determined.

The specific steps of the algorithm are provided in Algorithm 1.

The flow chart is shown below.

Figure 2 shows the overall flow of the algorithm. Initial abnormal detection is carried out on the input data stream through the sliding window in conjunction with ARIMA model.

Figure 3 shows the anomaly verification mechanism of specific process, and batch the historical data collected before. The first is to delete duplicates, dimension reduction, and sorting operation on batch data. After that, the data are clustered by the K-Means++ model. Then, original time series are converted into a Markov chain by using the clustering results. Finally, one step transfer matrix and the transition probability can be calculated. Through the principle of maximum and minimum, the hydrological sequence whose anomaly probability is higher than the threshold value is judged as the true outlier.

4. Results and Discussion

4.1. Experimental Environment and Dataset. The runtime environment is deployed in a cluster using four PCs, and the

```
Input: hydrological time series $X$, reliability $P$, sliding window size $L$, historical batch data $H$.
Output: the outliers in hydrological time series.
Step 1: clean the sequence $H$, including descending dimension, deleting duplicate value, sifting, and sorting
Step 2: using the value of $L$ as the initial starting position of the sliding window of $X$, the value of the $x_{L+1}$ is predicted, and as the window slides, the predicted value gradually forms a new time series $\{x_{L+1}, x_{L+2}, \ldots\}$
Step 3: the 95% confidence interval of the new time series is calculated and compared with $X$, the time point which is not in the confidence interval is obtained and get the exception point set $\{e_1, e_2, \ldots\}$
Step 4: taking historical data $H$ as input and training and establishing $K$-Means++ model, obtain the discrete state sequence $\{T_1, T_2, \ldots\}$
Step 5: compute the state transition matrix of the discrete state sequence
Step 6: the $K$-Mean++ model that is obtained in Step 4 is used for the exception point set of Step 3 and the value of its previous moment to obtain the state of anomaly and its previous moment
Step 7: estimate the value of the exception and its previous moment in Step 6 by state transition data frame and then output the confidence score
Step 8: repeat the above steps until no new data are entered
```

Algorithm 1: Hydrological time series anomaly detection algorithm.

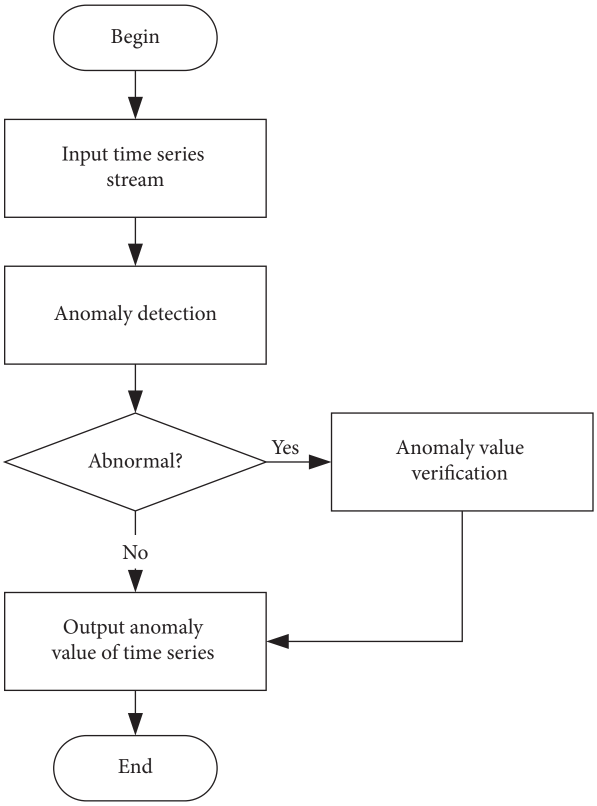

Figure 2: The whole process of anomaly detection algorithm.

hardware environment is as follows: CPU is Intel (R) Xeon (R) CPU E5645@2.40 GHz dual-core 24 CPU; memory is Kingston DDR3 1333 MHz 8G and 500 GB SSD flash memory. Operating system tools are Ubuntu 16.04 64-bit and Linux 3.11.0 kernel. The relevant software versions are as follows: Java 1.8 and Flink 1.5.2.

The experimental data are from the monitoring data from 2015 to 2017 for more than 70 hydrological stations located on Chu river, with a data size of 18910864 rows.

4.2. Forecast Testing. Since it is difficult to describe the flow data directly in the form of a graph, Figure 4 shows the change in water level in a whole certain period of time. Because of the time format of this experiment, the horizontal mark overlaps and the observation effect is poor. So, the number of days is used as the horizontal axis. Then, we introduce a time-continuous data stream for anomaly detection.

From Figure 4, there are indeed data in the data that deviates significantly from its neighbors' node, which is the

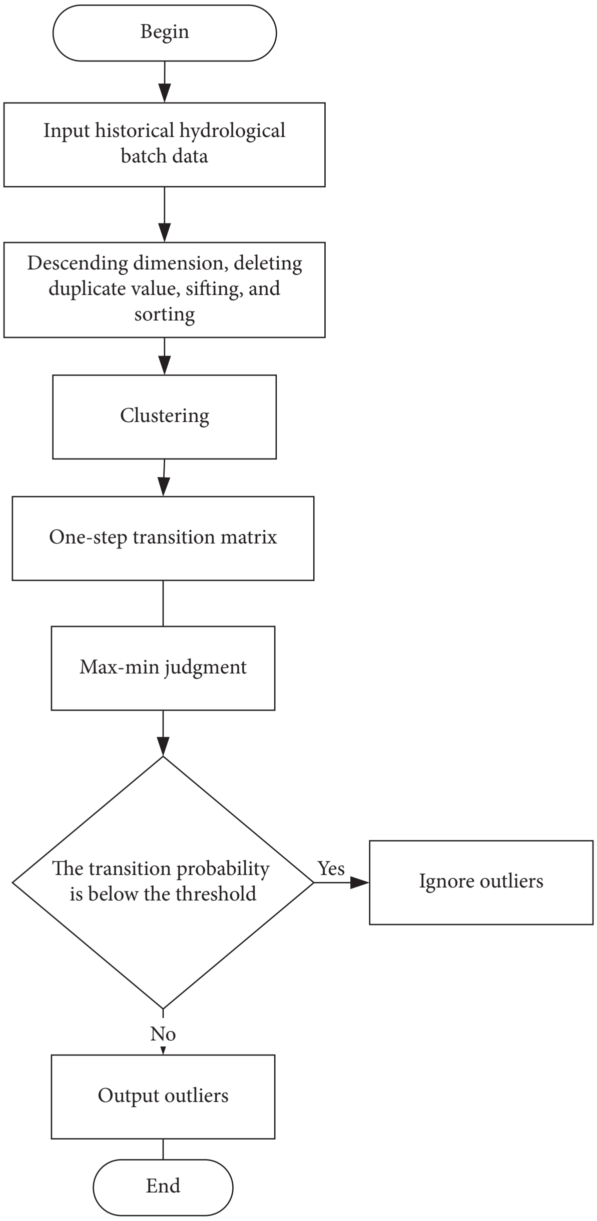
Begin
Input historical hydrological batch data
Descending dimension, deleting duplicate value, sifting, and sorting
Clustering
One-step transition matrix
Max-min judgment
The transition probability is below the threshold
Yes
Ignore outliers
No
Output outliers
End

Figure 3: Abnormal check mechanism flow chart.

anomaly we are trying to detect. The test results are given in Figure 5:

Figure 5 shows the measured values, predictive values, and confidence intervals with a confidence level of 95% in the case of a sliding window with a length of 6 and an anomaly detected on a given dataset when the reliability is 95%. From Figure 5, most of the points are very close to the normal value, but there are some points outside the interval, so they are judged as the suspect anomaly point.

4.3. Batch Data Clean. Before the anomaly detection, the data obtained should be cleaned and the data before cleaning are as follows.

We can see from Table 1, there are many problems in the original data, such as duplication, sort confusion, date format not conforming to data mining requirement, and existence of unrelated series. To solve the above problems, we have cleaned the initial 18910864 rows hydrological data based on Flink and compared with the cleaning time in traditional single mode, and the results are as follows.

As shown in Figure 6, when 15 stations are selected, the running speed of double node is slower than that of single node. However, with the increase in data size, the result calculated by double node has a small fluctuation, while the time of single node is significantly increased. Some Flink data after speed cleaning are shown in Table 2.

Table 2 retains important data, eliminates invalid data and duplicate data, and unifies the data format. These data play a key role in the later abnormal check.

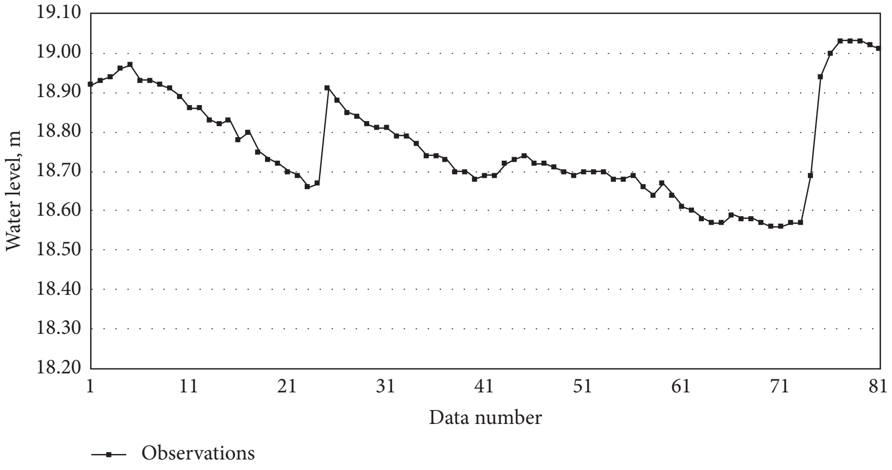

Figure 4: The part of initial hydrologic time series.

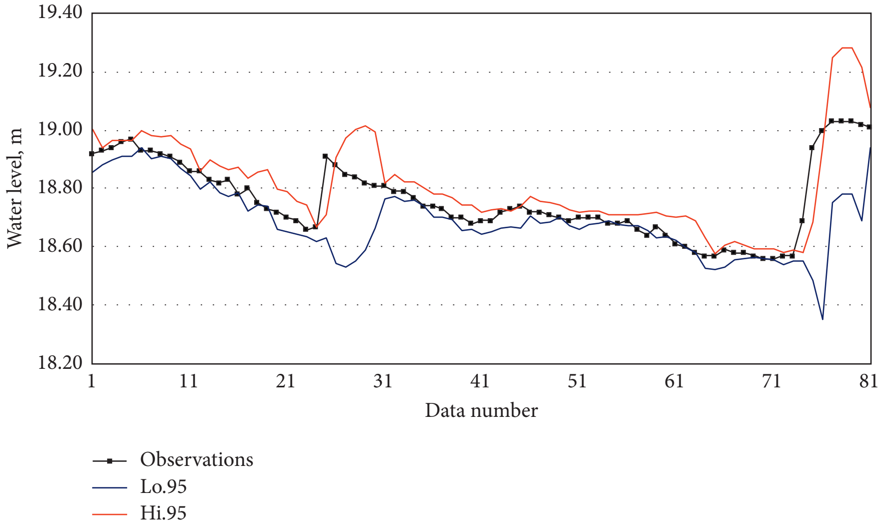

Figure 5: The part of exception detection results.

Table 1: Data before cleaning.

Station ID	Time	The water level	Source
12910520	27/4/2016 19:45:00	36.780	Null
12910520	27/4/2016 19:45:00	36.780	Null
12910560	27/4/2016 19:45:00	29.600	Null
12910580	27/4/2016 19:40:00	33.010	Null
60403200	27/4/2016 19:35:00	5.940	Nj34

4.4. Anomaly Verification Model Runtime. This experiment aims at the problem of that if predicting time series based on sliding window, high delay, and long running calculation time will appear. Flink is adopted for calculation, and the time of using Flink to execute the algorithm under different computing resources is compared. The results are as follows.

As can be seen from Figure 7, with the data of 15 and 35 hydrological stations selected, the running speed of the double node is not ideal, but when the data size rises to the level of tens of millions, the advantages of the double node can be reflected with the advantages of time growth speed and calculation time. It can be seen that, under the larger dataset, the double-node operation speed is faster, with a decrease in 17.43% in the fastest case. The relevant experimental results for the delay will be combined with the algorithm effectiveness experiment in the last part of the experiment.

4.5. Outliers Assessment. After anomaly detection, it is necessary to evaluate the detected outliers and use the state transition matrix to calculate the probability of its true anomaly value. The display section in the form of a data frame is as follows.

From Table 3, t_0 represents the state before the transfer and t_1 represents the state after the transfer. By classifying the initial hydrological time series, selecting the state of the anomaly value and their previous moments and looking in Table 3 to get the probability of the real anomaly value, some of the results are as follows.

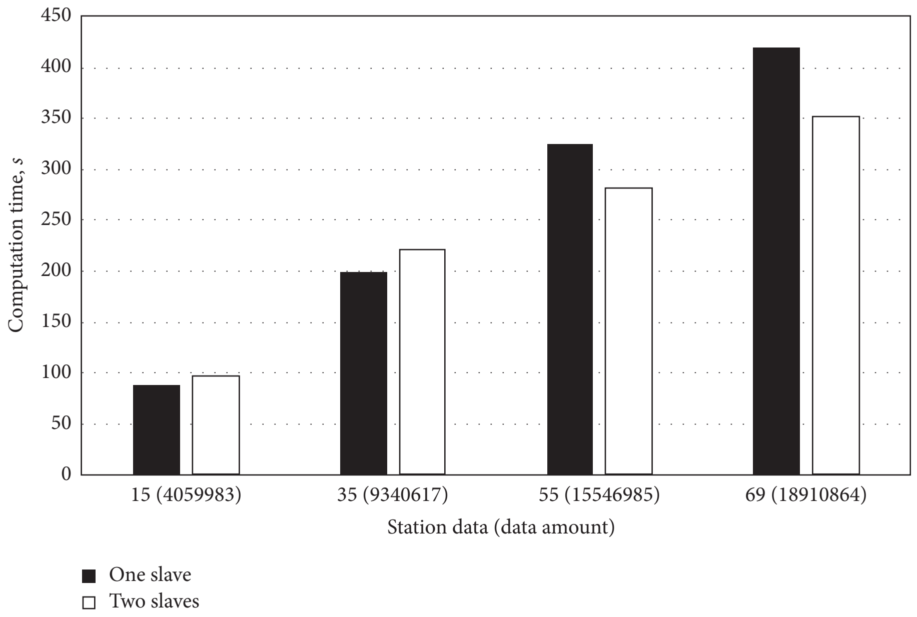

Figure 6: Comparison of running time in Flink in different slaves.

Table 2: Data after cleaning.

Station ID	Time	The water level
12910280	2016-01-29 14:15:00	22.3
12910280	2016-01-29 14:20:00	22.3
12910280	2016-01-29 14:25:00	22.3
12910280	2016-01-29 14:30:00	22.3
12910280	2016-01-29 14:35:00	22.3

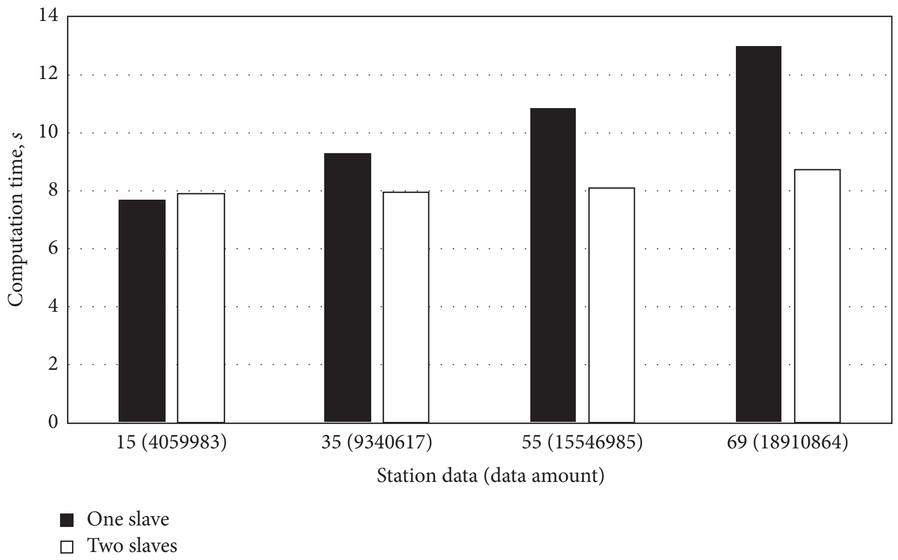

Figure 7: Comparison of Flink running time in different slaves.

Table 4 shows that the value detected by the experiment in *Forecast Testing* section is really the probability of abnormal value, it can be seen that the probability of real abnormal value for some detected value is 0, so we will remove these values whose probability of real abnormal value is less than 50% from the detected values.

4.6. Effectiveness and Accuracy. After anomaly detection, it is necessary to evaluate the detected outliers and use the state transition matrix to calculate the probability of its true anomaly value. In order to verify the validity and accuracy of this mechanism, this paper divides the experimental results into 4 categories. The first category is TP

Table 3: State transition data frame.

t_0	t_1	Probability
1	1	0.988283538
1	10	0.000000000
1	11	0.000000000
1	12	0.001054482
1	2	0.000000000

Table 4: Anomaly evaluation.

Station ID	Time	The water level	Pre_rz	Pro_ex
12910520	27/4/2016 19:45:00	36.780	22.31	98.270
12910520	27/4/2016 19:45:00	36.780	22.32	0.0000
12910560	27/4/2016 19:45:00	29.600	22.32	99.364
12910580	27/4/2016 19:40:00	33.010	22.32	99.364
60403200	27/4/2016 19:35:00	5.940	22.32	99.364

Table 5: Comparison of the sensitivity and specificity on the ARIMA model based on sliding window, MARS, and the algorithm in this paper.

Window size	Node number	Proposed algorithm		MARS		ARIMA based on slide window	
		Specificity (%)	Sensitivity (%)	Specificity (%)	Sensitivity (%)	Specificity (%)	Sensitivity (%)
10	1 node	82.16	80.98	87.44	82.55	81.57	54.55
	2 nodes	82.63	81.43	87.69	83.12	82.61	57.57
	3 nodes	82.41	80.84	87.57	82.51	81.44	55.82
30	1 node	94.74	90.81	94.99	87.84	93.12	67.28
	2 nodes	94.77	90.67	94.56	87.77	94.34	68.16
	3 nodes	93.98	90.98	94.73	87.98	93.45	68.45
50	1 node	99.46	92.40	97.65	90.41	99.13	72.06
	2 nodes	99.32	92.02	98.04	90.33	99.20	72.51
	3 nodes	99.61	92.98	98.11	90.63	99.29	72.91

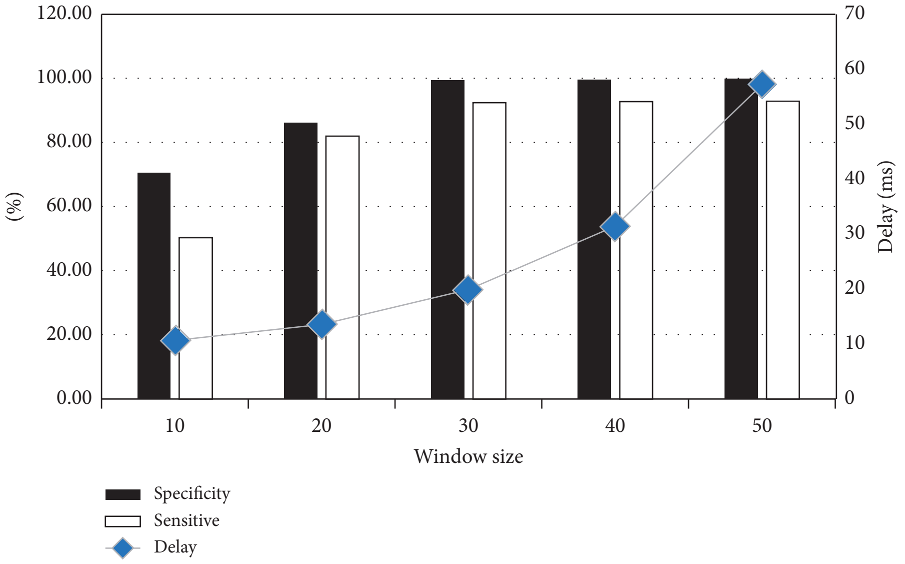

Figure 8: Effects of window size on specificity, sensitivity, and latency.

(true positive), and the actual abnormal is judged to be abnormal; the second category is FN (false negative), and the actual abnormal is judged to be normal; the third category is FP (false positive), and the actual normal is judged to be abnormal; the last Category is TN (true negative), and the actual normal is judged to be normal. TP and TN are ideal situations where FN and FP are not desired. This paper defines sensitivity, sensitivity = TP / (TP + FP), and specificity, specificity = TN/ (TN + FN). In this paper, the ARIMA model based on sliding window, MARS, and the algorithm in this paper are compared with the same hydrological time series, and the results are as follows (Table 5).

Table 5 shows the optimal results of the three models for comparison. The results show that the traditional window method is not significantly different from the method proposed in this paper in terms of specificity, maintaining high accuracy. In terms of sensitivity, as a nonlinear prediction model, MARS performs better in the case of a small sample, reaching a maximum of 83.12%. However, with the increase in the size of window, the increase is not as good as the algorithm proposed in this paper because the algorithm in this paper includes a historical verification mechanism, which uses historical data to verify real-time data. With the increase in window size, the algorithm proposed in this paper has a significant improvement compared with the traditional algorithm, with the sensitivity increasing from 72.91% to 92.98%. Comparing with the MARS model, the improvement is up to 2.75%.

As can be seen from Figure 8, with the increase in the number of windows, the specificity and sensitivity increased rapidly when the window size is about 30. During the window size from 10 to 30, the specificity increased by 29.47%, reaching 99.47%, the sensitivity increased by 42.08%, reaching 92.44%, and the delay increased by 93.32%, reaching 19.92 ms. However, when the window size is larger than 30, the specificity and sensitivity did not change much, but the delay rate increased to 57.21 ms and increased by 187.2%. Therefore, it is an ideal choice to set the window length to about 30, which can make the average delay less than 20 ms.

5. Conclusions

In the era of big data, traditional detection algorithms cannot meet the current needs. Based on the characteristics of the sliding window and the defects of traditional sliding window inspection, such as high time complexity and high error detection rate disadvantages, this paper puts forward a kind of hydrological time series anomaly detection method based on Flink. By using Flink calculation, this method reduces the computing time and combines two processes. After cleaning the data, the sliding window and the ARIMA model are used to forecast on the Flink platform. Then, the confidence interval is calculated for the predicted result and evaluated it as an anomaly value outside the interval range. Based on the detection result, the K-means algorithm is used to cluster the original data and the state transition probability is calculated.

Taking the data of hydrologic sensor obtained from the Chu River as an example of experimental data, experiments on the detection time and validity of outliers are carried out, respectively. The result shows that the million data using 2 slaves cost more time than 1 slave in the calculation time, but when the tens data are calculated, 2 slaves are better than 1 slave, and the maximum is reduced by 17.43%. The sensitivity of the evaluation is increased from 72.91% to 92.98%. In terms of delay, the average delay of different slaves is roughly the same, which is maintained within 20 ms. It shows that Flink can effectively improve the calculation efficiency by adding nodes when the method is used to detect tens of millions of hydrologic data. At the same time, the sensitivity of the method is significantly improved compared with the traditional method.

However, there is still room for further improvement in the detection accuracy of the algorithm proposed in this paper. The follow-up work will focus on more accurate identification of which are outliers and which are normal fluctuations caused by natural factors.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

The research work was done by two teams. The corresponding author QingHua Liu and Zihao Liu are from Jiangsu University of Science and Technology and Feng Ye and Professors Zhijian Wang are from Hohai University. The corresponding author was acting as the coordinator of the entire research effort.

Acknowledgments

This work was partly supported by the Fundamental Research Funds for the Central Universities, 2018 (B200202185); Jiangsu Province Key Research and Development Program (Modern Agriculture) Project under Grant no. BE2018301, 2017; Jiangsu Province Postdoctoral Research Funding Project under Grant no. 1701020C, 2017; and Six Talent Peaks Endorsement Project of Jiangsu under Grant no. XYDXX-078.

References

[1] H. Lu, W. T. Crow, Y. Zhu, Z. Yu, and J. Sun, “The impact of assumed error variances on surface soil moisture and snow depth hydrologic data assimilation,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 11, pp. 5116–5129, 2015.

[2] Y. F. Sang, Z. Wang, and C. M. Liu, “Research progress on the time series analysis methods in hydrology,” *Progress in Geography*, vol. 32, no. 1, pp. 20–30, 2013.

[3] J. S. Sun, Y. S. Lou, and Y. J. Chen, “Outlier detection of hydrological time series based on ARIMA-SVR model,” *Computer & Digital Engineering*, no. 2, pp. 225–230, 2018.

[4] Y. Yu, Y. L. Zhu, and D. S. Wan, “Time series outlier detection based on sliding window prediction,” *Journal of Computer Applications*, vol. 34, no. 8, pp. 2217–2220, 2014.

[5] D. M. Hawkins, *Identify of Outliers*, Chapman and Hall, London, UK, 1980.

[6] L. X. Niu, Z. F. Wang, and C. Z. Zang, “Hybrid model based on wavelet and ARIMA for short-term electricity price

forecasting," *Application Research of Computers*, vol. 31, no. 3, pp. 688–691, 2014.

[7] P. Gil, H. Martins, A. Cardoso, and L. Palma, "Outliers detection in non-stationary time-series: support vector machine versus principal component analysis," in *Proceedings of the 2016 12th IEEE International Conference on Control and Automation (ICCA)*, pp. 701–706, Kathmandu, Nepal, 2016.

[8] N. D. K. Vy and D. T. Anh, *Detecting Variable Length Anomaly Patterns in Time Series Data International Conference on Data Mining and Big Data*, pp. 279–287, Springer, Berlin, Germany, 2016.

[9] S. Ali, G. Wang, R. L. Cottrell, and T. Anwar, "Detecting anomalies from end-to-end internet performance measurements (PingER) using cluster based local outlier factor," in *Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pp. 982–989, Guangzhou, China, December 2017.

[10] H. C. Mandhare and S. R. Idate, "A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques," in *Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 931–935, Madurai, India, June 2017.

[11] Twitter/AnomalyDetection, 2019, https://github.com/twitter/AnomalyDetection?utm_medium=hao.caibaojian.com&utm_source=hao.caibaojian.com.

[12] Z. Y. Yang, Y. L. Zhu, and D. S. Wan, "Research on time series anomaly detection based on knowledge," *Granularity Computer Technology and Development*, vol. 26, no. 7, pp. 51–54, 2016.

[13] X. M. Liu and Y. R. Wang, "Anomaly pattern detection in time series based on outlier factor," *Computer Technology and Development*, vol. 28, no. 3, pp. 93–96, 2018.

[14] S. Zeng, S.-M. Chen, and M. O. Teng, "Fuzzy forecasting based on linear combinations of independent variables, subtractive clustering algorithm and artificial bee colony algorithm," *Information Sciences*, vol. 484, pp. 350–366, 2019.

[15] S. Zeng, S.-M. Chen, and K.-Y. Fan, "Interval-valued intuitionistic fuzzy multiple attribute decision making based on nonlinear programming methodology and TOPSIS method," *Information Sciences*, vol. 506, pp. 424–442, 2020.

[16] S. Chatterjee, S. Paladhi, S. Hore, and N. Dey, "Counting all possible simple paths using artificial cell division mechanism for directed acyclic graphs," in *Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1874–1879, New Delhi, India, 2015.

[17] D. Battré, S. Ewen, F. Hueske, O. Kao, V. Markl, and D. Warneke, "Nephele/PACTs," in *Proceedings of the 1st ACM symposium on Cloud computing—SoCC '10*, pp. 119–130, 2010.