Eng Int Syst (2022) 3: 233–242
© 2022 CRL Publishing Ltd

Engineering
Intelligent Systems

Similar Sequential Data Search Algorithm Based on Dimension-By-Dimension Strategy

Ying Liu[1,*] **and Hong Pan**[2]

[1] *College of Science, North China University of Science and Technology, Tangshan 063000, China*
[2] *ChangJiang Nanjing Waterway Bureau, ChangJiang Waterway Bureau, Nanjing 210011, China*

Because the traditional similar sequential data search algorithm considers only one-dimensional data, its data search accuracy is low, and the search data is not comprehensive. Hence, a similar sequential data search algorithm based on the dimension-by-dimension strategy is proposed. The algorithm measures the similar time series data in order to fill in the missing data in a time series, searches similar sub-sequences in time series data based on the strategy of dimension-by-dimension according to the data measurement results, gives a similarity threshold, queries the dynamic time-bending distance between sequences and the starting position of sub-sequences and candidate sub-sequences according to the threshold, and obtains the data by dynamically adjusting the search target hierarchical matching and search tasks. The experimental results show that under the influence of different levels of interference data, compared with the traditional search algorithm, the search matching accuracy of the proposed search algorithm is maintained at a relatively high level, the data search can obtain 89% of the expected search amount. Moreover, the process takes less time, demonstrating that the performance of the algorithm is superior, and can meet the current search requirements of similar time series data.

Keywords: Dimension-by-dimension strategy, similar time series data, search algorithm, similarity threshold.

1. INTRODUCTION

In regard to time series mining, the similarity of time series is the most basic and important problem. A similarity search of time series data involves the query of time series with similar change trends in time series data set according to existing data. It is an important method used to analyse time series. The similarity of sequences is not only directly related to time series clustering and similarity search, but also provides basic tools and a means of pattern discovery, data segmentation and other tasks. At the same time, it also provides a reference and basis for prediction and analysis of time series data, which has potential broad application in intelligent policy-making (Akshay et al., 2018; Hannan et al., 2018). By developing an efficient time series data search algorithm, users can find the similarity relationship between big data, which will greatly improve the application value of a time series database. Therefore, the similarity problem of time series is an important problem that requires an urgent solution (Loadsman et al., 2017; Guido et al., 2018).

Liu et al. (2018), a kind of classified group index method for sorting and searching efficient encrypted cloud data is proposed. After classifying the data, the key words are extracted according to the class to establish the group index, and the encryption time of index and query request is shortened by using several low-dimensional encryption keys instead of high-dimensional encryption keys (Liu et al., 2019). In addition, each group vector of the group index method corresponds to different categories, which not only updates the classification to improve the flexibility of updating documents, but also generates targeted trapdoors in the retrieval process, which improves the speed and efficiency of the search. Although theoretical and

[*] Corresponding Author Email: ncstly@126.com

experimental analysis shows that the method is feasible and effective, the data search results are incomplete. Yang et al. (2018) proposes a fast multi-keyword semantic ranking search method for data privacy in cloud computing. Firstly, the concept of domain weighted score is introduced in document scoring to distinguish different weights given by keywords (Jobaneh et al., 2019). Secondly, the semantic extension of search keywords, semantic similarity calculation, semantic similarity, domain weighted score and correlation score are combined to construct a more accurate document index. Finally, by matching the document marking vector and the query marking vector, a large number of irrelevant documents are filtered, which reduces the time required to calculate the document correlation score and document sorting, and improves the search efficiency. The experimental results show that this method can achieve fast sorting and improves the retrieval efficiency, but the accuracy of the data search presents problems. Zou et al. (2018), a query optimization method based on the Greenplum database is proposed, and an effective cost model is designed to estimate the query cost. Then, the parallel maximum minimum ant colony algorithm is used to search the connection sequence with the minimum query cost, that is, the optimal connection sequence. Finally, the optimal query plan is obtained according to the Greenplum database. The experimental results show that although this method can effectively search out the required data, the accuracy of the data search is not high enough.

In order to address the problem of single-dimension and poor search results obtained by the traditional similar time series data search algorithm, an algorithm based on the dimension-by-dimension strategy is proposed. In order to resolve the problems associated with the traditional algorithm, the algorithm is intended to increase the accuracy of search results, and to provide complete technical support for similar time series data search of massive data.

2. SEARCH ALGORITHM OF SIMILAR TIME SERIES DATA BASED ON DIMENSION-BY-DIMENSION STRATEGY

2.1 Measure Similar Time Series Data

Time series data has the characteristics of high dimension, high noise, high complexity and instability. If the original data is analyzed directly without preprocessing, the accuracy and reliability cannot be guaranteed, and it is not conducive to the efficiency of calculation and storage. Therefore, it is necessary to preprocess the original sequence before data analysis. The preprocessing methods mainly include filling the missing time points, smoothing the sequence, and normalizing the sequence. Time series refers to the sequence of variables arranged in sequence with time. Each time point can be a specific real value or a symbol for a specific pattern. The following is the definition of time series. Given time series X, there are:

$$X = x_1, x_2, \ldots, x_{|X|} \tag{1}$$

In the formula, $X(i) = x_i$, indicates that on the sequence X, the value at the sampling time point i is x_i, where x_i is a multi-dimensional value. Let $X(g, f)$ denote the subsequence formed by sampling from the sequence X, starting from the g time point and ending at the f time point:

$$X(g, f) = \frac{x_g, x_{g+1}, \ldots, x_{f-1}, x_f}{X} \tag{2}$$

Time series data is generally obtained from sensors and other devices, which may cause discontinuity or defect of source data due to faults and other reasons. In order to prevent the time axis offset caused by data loss, the missing data needs to be obtained before the analysis of the time series data. According to Equation (1) and Equation (2) above, a new time series is obtained with:

$$X_1 = \frac{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{|X|}}{X \cdot X(g, f)} \tag{3}$$

The data of the i time point is missing in the formula, that is, the data x_i is missing in Equation (3). For such data, the current mainstream processing schemes are as follows: through calculation, the missing time series data are discarded directly; the missing data points are calculated. In order to ensure the integrity and reliability of data, the statistical method is used to calculate and fill the missing data. There are two types of filling: the first is local mean filling where x_i is taken as the mean value within radius r, and there are:

$$x_i = \sum_{j=i-r}^{j=i+r} X_1(j)/2r \tag{4}$$

In the formula, r represents the effective radius; j represents the node coordinates of data missing location; $X_1(j)$ represents the time series of node j. The second is local high frequency value filling. x_i' is taken as the value with the highest frequency nearby. This is calculated with:

$$x_i' = \sum_{i}^{i} X_1(j)/p \tag{5}$$

In the formula: p represents the frequency of occurrence of the highest value near x_i. By filling the missing data, the sequence is smoothed. Known time series data are generally obtained from sensor equipment, which is often subject to interference by external factors, resulting in noise and a large number of random fluctuations. In order to eliminate the interference caused by these factors, it is generally necessary to smooth the data. According to Equation (5), the data weighting in the local range is used as the mathematical model of time series data smoothing:

$$S = \sum_{r=1}^{r=|X|} x_i' \cdot x_r \tag{6}$$

In the formula: x_r represents the time series data when the effective radius is r; β represents a attenuation function, $0 \leq \beta(n) \leq 1$, which represents the influence weight of other time points on the current value in the time series. The farther away the data is from x_i, the smaller is the impact on the smoothed x_i. For example, when the sequence X is missing, take x_i

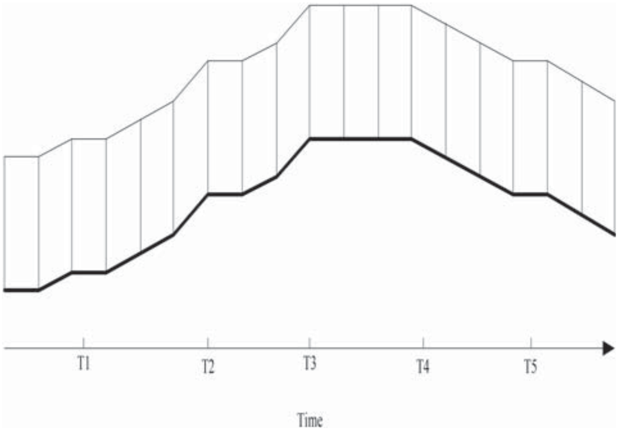


Figure 1 Euclidean distance between sequences.

for $\beta(0)=\frac{1}{2}$, $\beta(1)=\frac{1}{3}$, $\beta(2)=\frac{1}{6}$, then the smoothed x_i value is:

$$x_i=\frac{\frac{1}{6}x_1+\frac{1}{3}x_2+\frac{1}{2}x_3+\frac{1}{3}x_4+\frac{1}{6}x_5}{S-\beta(|i-r|)x_r} \tag{7}$$

In the same way, the smoothed time series data is obtained and normalized. It is known that different time series data often have different amplitude, period and absolute value, which most similar search algorithms find it difficult to distinguish. In order to eliminate the influence of these differences between time series data on subsequent analysis, the following equation can be used to normalize the data:

$$\begin{cases} x_i=\frac{x_i-\min(X)}{\max(X)-\min(X)} \\ x_i=\frac{x_i-mean(X)}{std(X)} \end{cases} \tag{8}$$

In the formula: $\min(X)$ represents the minimum fluctuation of time series data; $\max(X)$ represents the maximum fluctuation of time series data; $mean(X)$ represents the mean fluctuation of time series data; $std(X)$ represents the standard deviation. According to the results obtained by Equation (8), similar time series data are measured.

The similarity measurement of time series data is an important sub-topic in time series data mining research, and also the basis of time series data mining. The probability of having two identical time series data is very small, so it is necessary to measure the degree of similarity between two sequences by means of a similarity measure or a distance measure function. However, due to the complexity of time series data itself, there are often translation, scaling, discontinuity, nonlinear drift and mixed deformation along the time axis in the data. In order to mitigate as much as possible the impact of these problems on similarity measurement, and improve the efficiency of time series similarity measurement, the measurement of similar time series data is completed according to the distance measurement function. The commonly-used distance measurement functions include Minkowski distance, Euclidean distance, DTW distance, LCS distance and cosine similarity. In order to ensure the reliability of the search algorithm, Euclidean distance is used to measure the similarity.

Euclidean distance (ED) is the most widely-used and most simple distance calculation method in sequence measurement. Euclidean distance considers sequence X (length n) as a point in Euclidean space of dimension n, and the coordinate value of this point is the value of sequence X, so the Euclidean distance of two sequences is the space distance between two points in n dimensional space. It is known that the length of two sequences X and Z is $|X|$ and $|Z|$ respectively. When $|X|=|Z|$, the Euclidean distance between them is:

$$f(X,Z)=\sqrt{\sum_{i=1}^{|X|}(x_i-z_i)^2} \tag{9}$$

It is simple to calculate Euclidean distance, making it suitable for research domains that require highly efficient algorithms and a small fluctuation range. However, for the two sequences with different amplitudes or with offset or stretch along the time axis, even if their fluctuation trend is similar, the calculated distance may have a large deviation. For the two sequences with small measurement distance, the waveform difference between them could be large. This is due to the noise and volatility in time series data, so similar sequences will take on many forms, such as noise interference, time axis translation, time axis expansion, nonlinear drift and data point discontinuity, as shown in Figure 1.

On the whole, the two sequences in Figure 1 are similar, with only a few offsets and stretches along the time axis. When Euclidean distance is used for measurement, because it is only the difference of corresponding points in a simple linear accumulation sequence, and does not match the data with offset or stretch, a large distance value will be generated after measurement, which will affect the accuracy of subsequent analysis.

To solve these problems, an improved Euclidean distance algorithm is proposed. Before the Euclidean distance is calculated, the sequence is normalized in sections to eliminate the negative effects of the aforementioned deformation. In this method, the sequence is divided into several sub-sequences of equal length, and then each sub-sequence is normalized one by one, instead of the whole sequence. In this method, different weights are assigned to different positions of the sequence. First, the alignment sequence W is divided into several parts of equal length, and different weights are assigned to each part, namely:

$$W = w_1, w_2, \ldots, w_{|W|}, u_1, u_2, \ldots, u_{|W|} \qquad (10)$$

In the formula, W represents the alignment sequence; $w_{|W|}$ represents the alignment sequence data value with length $|W|$. According to Equations (9–10), the calculation formula of similarity distance between sequence X and comparison sequence W is obtained with:

$$f(X, W) = \sqrt{\sum_{i=1}^{|X|} \lambda_i (x_i - w_i)^2} \qquad (11)$$

In the above formula, λ_i represents the similarity weight corresponding to the i data. Through the above process, the preprocessing of time series data is completed, which provides the basis for more accurate measurement of similar time series data.

2.2 Searching Similar Sub-Sequences of Time Series Data Based on Dimension-by-Dimension Strategy

After the similarity measurement results are obtained, the time series data similar sub-sequences are searched based on the dimension-by-dimension strategy. With the dimension-by-dimension strategy, by introducing an adjustment factor, similar sub-sequence searches can be carried out in multiple associated time series data that have different dimensions and levels.

The dimension-by-dimension strategy originates from the iterative improvement strategy; that is, after one dimension of information for a current individual is updated by a formula, it will be combined with other dimension information for the current individual to form a new individual, and then the new individual is evaluated. Suppose the objective function $g(Y) = y_1^2 + y_2^2 + y_3^2$, and the i individual in the h generation is $Y_{h,i} = (0.5, 0.5, 0.5)$ [8]. For individual $Y_{h,i}$, randomly select the information of dimension j to update, and the default value is $j = 1$. Assuming that the dimension information is updated from 0.5 to 0.2, the dimension-by-dimension strategy combines the updated information and other dimension information to form a new individual $\Delta Y_{h,i} = (0.2, 0.5, 0.5)$, and then evaluates the individual. It should be noted that different selection strategies can be used to select new individuals and current individuals. In addition, the information in the dimension can be updated through the dimension policy using the following formula:

$$Y_{h+1,i} = \frac{\Delta Y_{h,i} + s(Y_{h,j} - Y_{h,i})}{f(X, W)} \qquad (12)$$

In the formula, s is the random number of interval $(-1, 1)$; $Y_{h,j}$ is the random individual of generation h. Similar sequence search is the basis of classification, clustering and anomaly detection in time series data mining. It is one of the main tasks of time series data mining and has important theoretical and practical value. Different from the normal database query, the similar sequence search problem involves finding the time series data which is only slightly different from the given query sequence. At present, there are two types of similar sequence search problems in the existing research: similar full sequence search and similar sub-sequence search. In the study of similar full sequence search, given a set M comprised of multiple sequential data, its goal is to find the sequence similar to the given query sequence N in M. In the study of similar sub-sequence search, given a longer sequential data V, its goal is to find the sub-sequence segment similar to the given query sequence Q in V (Rinku et al., 2018; Murtafi'ah et al., 2019).

A full sequence search is generally used in the mining research on multiple time series data sets. For example, in the clustering approach used for multiple time series data sets, it is necessary to obtain the degree of similarity between each time series data in the data set and several clustering center sequences. In this case, the related research results of a similar full sequence search can be used. Due to the large scale of time series data produced in many application fields, the overall processing is not very efficient, limiting the scope of application of similar full sequence search. The requirement of similar sub-sequence search is more common in application. The problem of similar sub-sequence search is extended to similar sub-sequence search, trying to find those sub-sequences which are similar to the query sequence from the given long-term series data. The relevant formal definitions are as follows:

Given a similarity threshold σ, a query sequence N and a multi-temporal data set M, if the distance between temporal data V and N in M is less than σ, that is, $E(V, N) \leq \sigma$, then N similarity full sequence search results for M include V, which is a specific description of similarity full sequence search. For similar sub-sequence search, given a similarity threshold σ, a query sequence N and a time series data V, if the distance between sub-sequence $V[t_g, t_f]$ and N of V is less than σ, there is inequality $E(V[t_g, t_f], N) \leq \sigma$, then the search results of similar sub-sequence N for V include $V[t_g, t_f]$ (Teng et al., 2018).

Specifically, Specifically, similar time series data sub-sequences are searched based on dimension-by-dimension strategy. According to given sequence $V = \{v_1, v_2, \ldots, v_m\}$ and sequence $N = \{n_1, n_2, \ldots, n_m\}$, the dynamic time bending distance $E(V[t_g, t_f], N)$ between sub-sequence $V[t_g, t_f]$ in sequence V and query sequence N, and the calculation formula of starting position $gv(t, i)$ between candidate sub-sequence $N[1, i]$ and query sub-sequence V are as follows:

$$\begin{cases} E(V[t_g, t_f], N) = Y_{h+1,i} \cdot g(t_f, m) \\ g(i, j) = dist(v_i, u_i) + dist_{best} \\ dist_{best} = \min \begin{cases} g(i, j-1) \\ g(i-1, j) \\ g(i-1, j-1) \end{cases} \\ g(i, 0) = 0, g(0, j) = \infty \end{cases} \qquad (13)$$

In the formula: $dist_{best}$ is the measurement result of similar time series data after the step size information is updated by dimension-by-dimension strategy. It can be seen from the formula above that in the process of sub sequence search, it is not necessary to recalculate the whole dynamic time bending distance matrix because of the use of matching initial position information. For each data in V, it is only necessary to

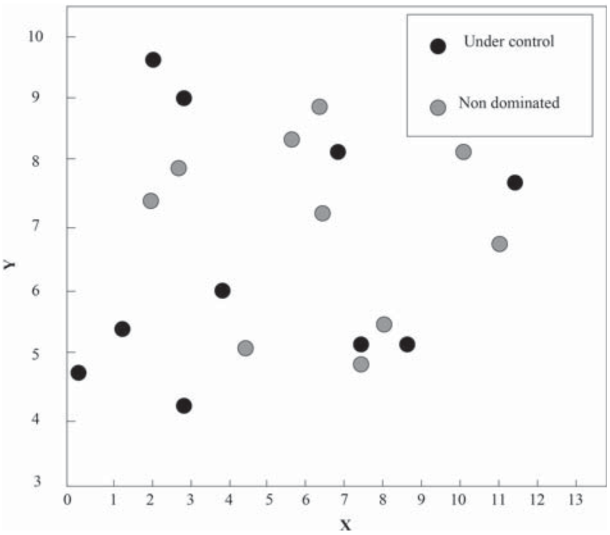


Figure 2 The dominance relationship between individuals.

calculate the dynamic time bending distance and the starting position of a column of elements in the matrix to solve the online sub sequence search (Zhang et al., 2018).

2.3 Dynamically Adjust the Free Search Target to Achieve The Hierarchical Matching of Search Algorithm

The application of the dimension-by-dimension strategy to search the similar sub-sequence of time series data may not produce the best results. Therefore, it is necessary to adjust the search target dynamically, obtain the key target according to the multi-objective comparison and optimization, and obtain the hierarchical matching of the search algorithm. Similar sequential data search algorithms need to achieve multiple different search objectives, so the multi-objective optimization method is adopted. A solution may optimize one of the functions, but it can also degrade the performance of other functions. Therefore, in the case of multi-objective optimization, each objective can be optimized only by trade-off or compromise, and all objectives can be optimized as much as possible (Jiang et al., 2018; Zhenilin et al., 2019). Its mathematical form is:

$$\begin{cases} \min b = f(a) = [f_1(a), f_2(a), \cdots, f_m(a)] \\ s.t.g_i(a) \leq 0, i = 1, 2, \dots, u \\ h_j(a) = 0, j = 1, 2, \dots, v \\ \hat{a} = (a_1, a_2, \dots, a_n) \in T, \hat{b} = (b_1, b_2, \dots, b_m) \\ T = \{(a_1, a_2, \dots, a_n) | \gamma_i \leq a_i \leq \kappa_i\} \\ \gamma = g(i, j)(\gamma_1, \gamma_2, \dots, \gamma_n), \kappa = (\kappa_1, \kappa_2, \dots, \kappa_n) \end{cases} \tag{14}$$

In the formula:, $\hat{a} \in T$, $\hat{a}$ is the decision objective space; T is the variable dimension formed by the decision objective space; $\hat{b} \in g(t_f, m)$ is the objective vector; m is the total number of objective functions; $g_i(a) \leq 0$ is the inequality constraint condition, the number is u; $h_j(a) = 0$ is the equality constraint condition, the number is v; γ and κ represents the boundary value of a_i. The appropriate solution for the multi-objective optimization problem is not limited to one, but is a set of optimal solutions that compromise all objective functions. Generally speaking, these optimal solutions can also be called non-dominated solutions, and the set of solutions is known as a non-dominated solution set. The key to dealing with the multi-objective optimization problem is to obtain the optimal set so that the appropriate solution can be selected according to the actual situation (Saer et al., 2018). The common basic concepts of an optimal solution set are:

If vector a_1 dominates a_2, it can be recorded as: $a_1 \prec a_2$, and satisfies: If $f_i(a_1) \leq f_i(a_2)$, $i = 1, 2, \dots, m$, then it exists $\exists i \in \{1, 2, \dots, m\}$, $f_i(a_1) < f_i(a_2)$. If a_1 is the optimal solution, then there is no other solution $a_2 \succ a_1$ in the feasible region. Figure 2 is a schematic diagram of the dominance relationship between two individuals in the target space.

Dynamically adjust the search target according to the relationship in Figure. Let the number of individuals in the population be m, and according to Equation (14), obtain the position of the j individual in the n dimension optimization search space, which is recorded as $B_j = (b_{p1}, b_{p2}, \dots, b_{pm})$, ic represents the current evolution times, and IC represents the total evolution number (Anna et al., 2018). In the search space, the fitness k_{cj} of multi-objective function of each individual small step search c can be defined as:

$$k_{cj} = \prod_{p=1}^{p} k_{cjp}^2, k_{cjp} - k_p(b_{cji})\gamma \tag{15}$$

In the formula: k_{cjp} is the corresponding fitness of the p objective function of individual j. In order to speed up the search success rate of the algorithm, the starting point of the individual's next search is improved. The position with a high k_{cjp} value is taken as a new search point. The new round of search points of individual k_{cj} are defined as:

$$\hat{k}_{cj} = k_{cjp}(k_{cjp} \in \max f(b_{ij}, b_{2j}, \dots, b_{cj})) \tag{16}$$

According to the new round of search points, the multiple time series patterns are divided to realize the hierarchical matching of the similar time series data search algorithm. There are many ways to express the pattern of multivariate time series, but the most direct way is to express the multivariate time series with the original data. This method can capture and describe all the features of the original sequence accurately, without missing information. However, in the face of a large number of multiple time series databases, people are often more concerned about whether they can grasp the overall shape characteristics of multiple time series first, rather than being concerned with a small detail. Therefore, according to the time series feature points, the multi-element time series is segmented and connected to these feature points, so that the main shape of the sequence can be maintained after transformation, and the original curve with a huge amount of information becomes several straight-line segments. Then the slope of each small segment is calculated. If the slope of this paragraph is greater than 0, use 1, otherwise use 0. In this way, the original complex multiple time series is transformed into a matrix containing only symbols, simplifying the complex problems, and achieving complete matching of similar time series data through hierarchical search (Aysegul et al., 2018).

According to the trend feature of the extracted sequence, the trend distance is calculated, and the sequence with a similar trend is obtained and then added to the second candidate set. In this rough trend similarity candidate set, the search range is greatly reduced. At this time, Euclidean distance is used to search the similarity of the original sequence. After the original complex multivariate time series is transformed into a symbol matrix containing only 0 and 1, similarity search is carried out in the symbol matrix set, which is a process involve the rough matching of sequence trends. During this process, we need to give the corresponding similarity measurement standard, that is, the definition of trend similarity. In this study, the definition of trend distance is improved in the original unitary time series, and it is extended to the definition of trend similarity of multivariate time series: For two equal length multivariate time series A and B, the character matrix P and Q are respectively obtained after symbol conversion of their sequences. If A and B meet the conditions:

$$T(A, B) = \hat{k}_{cj} - \sum_{i,j} |P_{i,j} - Q_{i,j}| \leq \sigma \tag{17}$$

The trends of multivariate time series A and B are similar. The result set after the rough matching is regarded as the candidate set for fine matching. For fine matching, Euclidean distance is used to measure the similarity distance between two original multivariate time series. This paper also uses the definition of similarity of unitary time series for reference, and extends it to the vertical moving similarity of multivariate time series. Let two equal time series be A^* and B^*. If A^* and B^* satisfy the following formula:

$$T(A^*, B^*) = \left[\sum_{i=1,j=1}^{n,m} \left[(b^*_{i,j} - a^*_{i,j}) - \left(\frac{1}{mn} \sum_{i=1,j=1}^{n-1,m-1} b^*_{i,j} - \frac{1}{mn} \sum_{i=1,j=1}^{n-1,m-1} a^*_{i,j} \right) \right]^2 \right]^{\frac{1}{2}} \leq \sigma^* \tag{18}$$

Assume that multiple time series A^* and B^* are similar in vertical movement. By calculating Euclidean distance and fine matching, the final similarity set can be selected from the candidate set to complete the similarity search task of multiple time series. The similar sequential data search algorithm based on dimension-by-dimension strategy is now realized.

3. EXPERIMENTAL STUDY

In order to test the reliability of the proposed similar time series data search algorithm based on the dimension-by-dimension strategy, a comparative experiment is conducted. The similar time series data search algorithm based on the dimension-by-dimension strategy is compared with the traditional Hannan et al. (2018), Jiang et al. (2018) and Jobaneh. (2019) method, and the results of the search matching using different methods for similar time series data is analyzed, specifically in terms of data search accuracy, data comprehensiveness. and data search time. The test results enable conclusions to be drawn about the effectiveness of each of the tested methods. For the experiment, MATLAB software is used to process the data.

3.1 Analysis of Experimental Data and Test Objects

The experiment was conducted using the steps described below.

Randomly select a node system as the test object; the key timing data nodes of the system are shown in Figure 3.

Connect the experimental test data to the nodes of the above system. Take a website as the test background. Match the system nodes, as shown in Table 1.

All website information is known to be connected with the access point information of the test system. The data with the largest per capita search volume obtained during 48 consecutive hours is used as the query sub-sequence. In total, five independent query sub-sequences are selected. One query sub-sequence is randomly selected for the real-time search of related data, and the other four query sub-sequences are used as alternative sequences. Figure 4 depicts the query diagram of this query sub-sequence.

According to the query curve in Figure 4, in different query stages, the amount of query data in query sub-sequences is constantly changing. It can be seen that the query sub-sequence meets the criteria for testing, and the experimental test can begin.

3.2 Experimental Results and Analysis

In order to ensure the reliability of the experimental test, under the two conditions of small amount of interference data and large amount of interference data as the test premise, different methods are used to carry out a search of similar timing data. Figure 5 below shows the data search results indicating the comparative accuracy of different methods when there is a small amount of interference data. The accuracy is expressed by a numerical value, specifically 0–1.0. The larger the

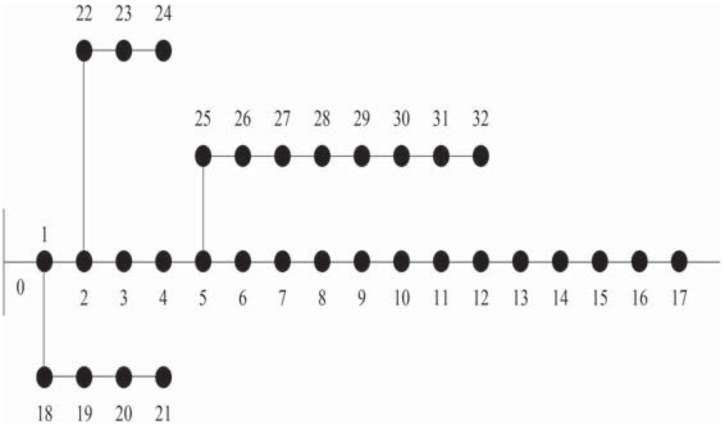


Figure 3 Node numbers of test system.

Table 1 Information access distribution of various websites to be selected.

Number	Website type	Access node
01	Clothing and accessories	13, 14, 15, 16, 17, 18, 23, 24, 29;
02	Food and beverage	6, 7, 19, 20, 21, 22;
03	Travel accommodation	1, 2, 3, 4, 5, 8, 9, 30, 31, 32;
04	Traffic	18, 21, 32;
05	Daily supplies	13, 15, 20, 29;
06	Electronics	2, 4, 9, 24, 30;
07	Leasing information	13, 15, 18, 20, 21, 29, 32;
08	Skin care makeup	10, 11, 12, 25, 26, 27, 28;

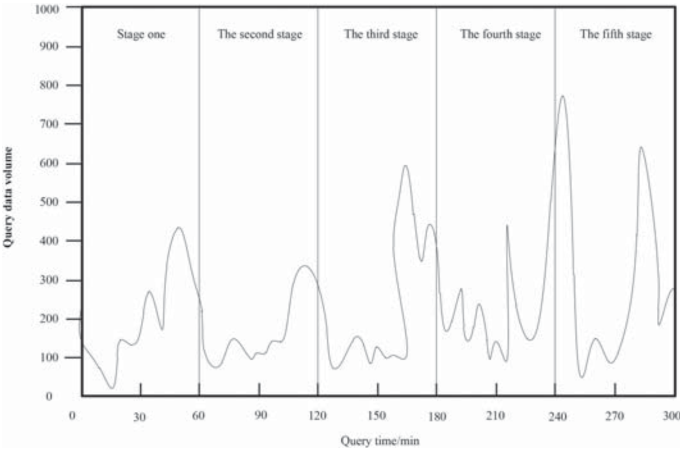


Figure 4 Query diagram of query sub-sequence in test.

numerical value, the higher is the accuracy of the search result.

As shown in Figure 5, when the key degree parameters of the search target become larger and larger, the data search accuracy of other methods, except for that of reference [5], generally shows a downward trend, but the data search accuracy of the algorithm proposed in this paper is significantly higher than the reference [5] method, the reference [6] method and the reference [7] method. The curve trend in the graph indicates that the accuracy of the search results for similar time series data of the proposed search algorithm has been kept above 0.5, while the search results for similar time series data obtained by the method in reference [7] are below 0.3. This is because, in order to eliminate the influence of different scales of time series data on the subsequent analysis, the algorithm in this paper normalizes the data and improves the accuracy of the data search.

The premise of the test is the large amount of interference data. The test results under different experimental conditions are shown in Figure 6.

The curve trend in Figure 6 indicates that the data search accuracy of the proposed algorithm decreases significantly only when the critical parameter of the search target is 0–0.1. Since the critical parameter of the search target is 0.1, there is no significant decline. However, the traditional data search method, due to the lack of data preprocessing, cannot remove any interference when the amount of data is large. Hence, the search algorithm based on dimension-by-dimension strategy is proven to be more rigorous.

In order to further verify the effectiveness of the algorithm

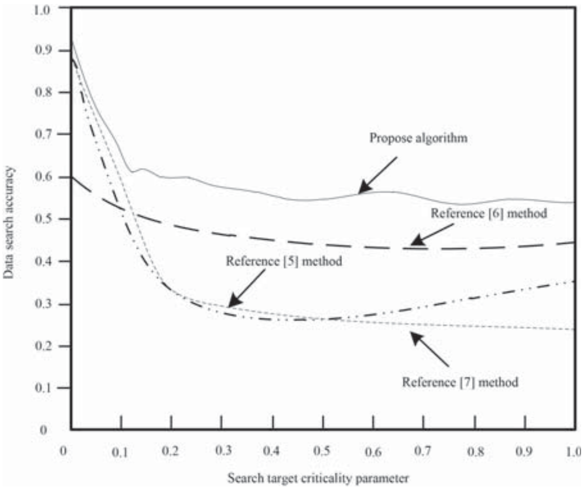


Figure 5 Experimental test results with small amount of interference data.

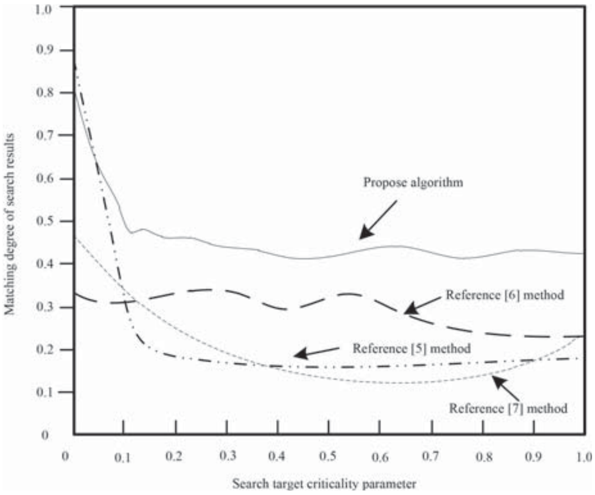


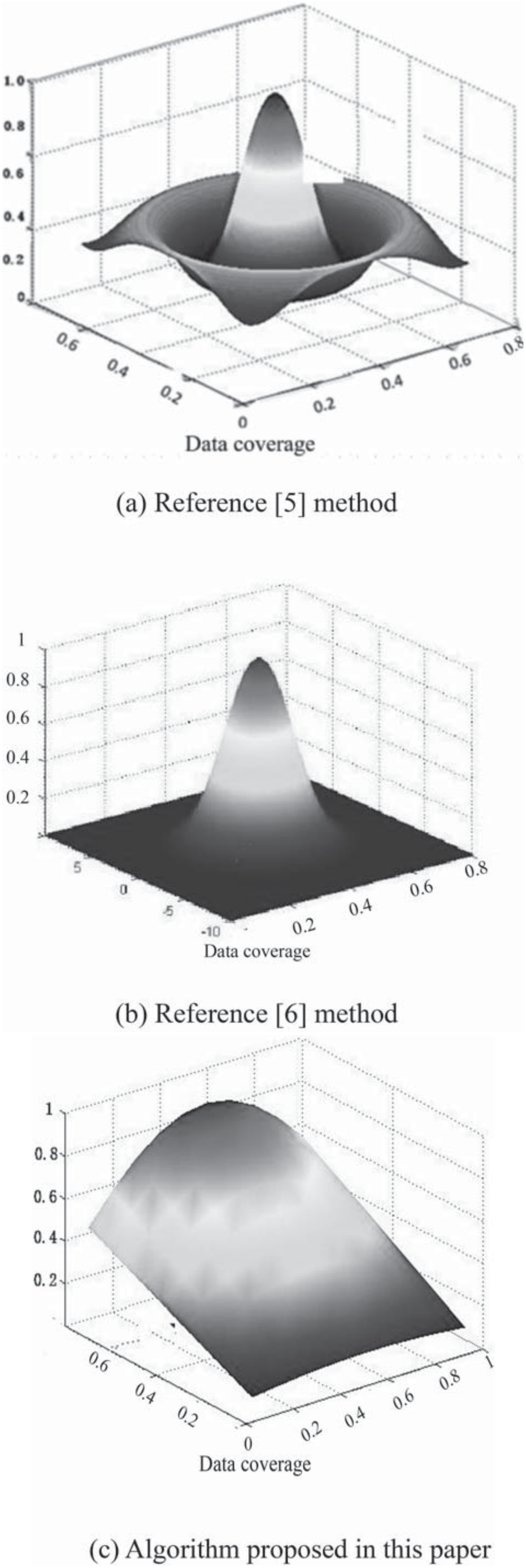
Figure 6 Experimental test results with large amount of interference data.

proposed in this paper, the comprehensiveness of the data search is taken as the index to compare the data search results of different algorithms. These results are shown in Figure 7.

The analysis of Figure 7 shows that the search results obtained by the proposed algorithm for similar time series data cover a wider range, while the coverage of reference [5] method and reference [6] method is obviously not as high as that of the proposed method, indicating that the data obtained by this algorithm is more comprehensive. This is because the algorithm does not consider that time series data is generally obtained from sensors and other devices. These devices may cause discontinuities or defects of source data due to faults and other reasons. In order to prevent data loss caused by time axis offset, the missing data is calculated before the analysis of time series data, thereby improving the coverage of the data search.

4. CONCLUSIONS

Similarity search is an important part of time series data mining. The similarity search of multiple time series is closely related to the definition of construction pattern.In this study, which concerns the hot research topic of similar time series data search, the pattern of sequence is defined from shape feature, and a similar time series data search algorithm based on dimension-by-dimension strategy is proposed. By extracting feature points, this method can effectively discover the shape features of time series. By means of hierarchical matching, the trend similarity is confirmed, followed by the confirmation of the similarity of details. Thus, the search space is gradually decreased, and the accuracy and comprehensiveness of the search are improved. Experimental results show that the algorithm reduces the amount of computation, improves the efficiency of algorithm execution, and improves the coverage and accuracy of the data search results. Although this paper introduces the idea of a dimension-by-dimension strategy and classification for time series similarity search, and has solved some problems through experiments, this theory and the experimental data results have limitations: whether they can deal well with other kinds of data, and have useful practical application, remains to be studied and discussed.



(a) Reference [5] method

(b) Reference [6] method

(c) Algorithm proposed in this paper

Figure 7 Comparison results of data search comprehensiveness of different methods.

REFERENCES

1. Akshay, K., Gauri, S. 2018. Quasi-oppositional harmony search algorithm based optimal dynamic load frequency control of a hybrid tidal–diesel power generation system. *IET Generation Transmission & Distribution, 12*(5), 1099–1108.
2. Anna, K., Janusz, B., Guilhem, M. 2018. Stochastic models in the DORIS position time series: Estimates for IDS contribution to ITRF2014. *Journal of Geodesy, 92*(7), 743–763.
3. Aysegul, O., Rahul, P., Troy, B., et al. 2018. Algorithm 990: Efficient atlasing and search of configuration spaces of point-sets constrained by distance intervals. *ACM Transactions on Mathematical Software, 44*(4), 1–30.
4. Guido, C., Reza, A. 2018. Power distribution network topology detection with time-series signature verification method. *IEEE Transactions on Power Systems, 33*(4), 3500–3509.
5. Hannan, M.A., Ali, J.A., Mohamed, A., et al. 2018. Quantum-behaved lightning search algorithm to improve indirect field-oriented fuzzy-PI control for IM drive. *IEEE Transactions on Industry Applications, 54*(4), 3793–3905.
6. Jiang, Y.L., Liu, M.N., Chen, T., et al. 2018. TDOA passive location based on cuckoo search algorithm. *Journal of Shanghai Jiaotong University (Science), 23*(3), 368–375.
7. Jobaneh H.H. 2019. An Ultra-Low-Power and Ultra-Low – Voltage 5 GHz Low Noise Amplifier Design with Precise Calculation. *Acta Electronica Malaysia, 3*(2), 23–30.
8. Liu, L.G., Sun, H., Jia, H.L., et al. 2019. CGIM: Classificatory group index method for efficient ranked search of encrypted cloud data. *Acta Electronica Sinica, 47*(2), 331–336.
9. Loadsman, J.A., McCulloch, T.J. 2017. Widening the search for suspect data – Is the flood of retractions about to become a tsunami? *Anaesthesia, 72*(8), 931.
10. Murtafi'ah B., Putro N.H.P.S. 2019. Digital Literacy in The English Curriculum: Models of Learning Activities. *Acta Informatica Malaysia, 3*(2), 10–13.
11. Rinku, R., Debdeep, S., Manjunatha, M., et al. 2018. A fingertip force prediction model for grasp patterns characterised from the chaotic behaviour of EEG. *Medical & Biological Engineering & Computing, 56*(4), 1–13.
12. Saer, S., Malcolm, J.R., Kine, B., et al. 2018. Combining a deconvolution and a universal library search algorithm for the nontarget analysis of data-independent acquisition mode liquid chromatography-high-resolution mass spectrometry results. *Environmental Science & Technology, 52*(8), 4694–4701.
13. Teng, M., Timothy, D.J., Farouk, S.N. 2018. Time series analysis of fMRI data: Spatial modelling and Bayesian computation. *Statistics in Medicine, 37*(2), 2753.
14. Wang, H.J., Wang, Y. 2018. Optimization retrieval simulation of user interest information in distributed database. *Computer Simulation, 35*(6), 428–431.
15. Yang, Y., Liu, J., Cai, S.W., et al. 2018. Fast multi-keyword semantic ranked search in cloud computing. *Chinese Journal of Computers, 41*(06), 1126–1139.
16. Zhang, J.J., Lin, G., Li, W.X., et al. 2018. An iterative local updating ensemble smoother for estimation and uncertainty assessment of hydrologic model parameters with multimodal distributions. *Water Resources Research, 54*(3), 1716–1733.
17. Zhenilin OU, Xie H.B., Yang X., Bojia P.I, Ciel R.D., Zhao X.R.A. 2019. A Look at Millennial Attitudes Toward Ai Utility in The Class. *Information Management and Computer Science, 2*(1), 07–09.
18. Zou, C.M., Xie, Y., Wu, P. 2018. Query optimization based on Greenplum database. *Journal of Computer Applications, 38*(2), 478–482.