JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **17**, Number **1**, January **2021**

doi:10.3934/jimo.2019107

pp. **205–220**

A BIDIRECTIONAL WEIGHTED BOUNDARY DISTANCE ALGORITHM FOR TIME SERIES SIMILARITY COMPUTATION BASED ON OPTIMIZED SLIDING WINDOW SIZE

Cheng Peng

School of Automation
Central South University, Changsha, Hunan410083, China
School of Computer
Hunan University of Technology, Zhuzhou, Hunan412007, China

Zhaohui Tang* and Weihua Gui

School of Automation
Central South University, Changsha, Hunan410083, China

Qing Chen and Jing He

School of Computer
Hunan University of Technology, Zhuzhou, Hunan412007, China

(Communicated by Biao Luo)

Abstract. The existing method of determining the size of the time series sliding window by empirical value exists some problems which should be solved urgently, such as when considering a large amount of information and high density of the original measurement data collected from industry equipment, the important information of the data cannot be maximally retained, and the calculation complexity is high. Therefore, by studying the effect of sliding window on time series similarity technology in practical application, an algorithm to determine the initial size of the sliding window is proposed. The upper and lower boundary curves with a higher fitting degree are constructed, and the trend weighting is introduced into the *LB_Hust* distance calculation method to reduce the difficulty of mathematical modeling and improve the efficiency of data similarity computation.

1. **Introduction.** Time series is a quantity that arranged in chronological order, changing with time and interrelated with each other, and the interval between series can be uncertain [3]. Time series analysis is one of the strong applied branches in probability statistics, computer, and other subjects, and is an important field of data analysis [5], such as similarity query [1], anomaly detection [4], clustering [27], state evaluation [13], trend analysis [29], etc. By this kind of analysis, the deep-level relations such as the implicit correlation in the time series can be revealed and the internal rules of movement, change, and development of objects also are discovered, which is of great practical significance for people to correctly understand

2020 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Time series, bidirectional boundary distance, sliding window,LBHust distance,similarity computation.

* Corresponding author: Zhaohui Tang.

things and make scientific decisions accordingly. Time series mining is one of the most challenging research problems in the field of data mining, the key step is to map the original data from a high-dimensional feature space to a low dimensional space to reduce the time complexity [14], the similarity clustering is an important research direction of time series mining, many other time sequence mining methods are based on it.

2. **Related studies.** In the research field of time series, various time series dimensionality reduction methods proposed by scholars are closely related to the sliding window. Generally, the time series are segmented by the sliding window size to reduce the computational complexity. Mainly includes the Piecewise Aggregate Approach(PAA) [19], Sliding Window Segment Representation (SWSR)with supporting investment decisions partition based on improved PAA [35], Symbolic Aggregate Approximation(SAA) [24] and Discrete Fourier Transform(DFT) [22], Dynamic Time Warping(DTW) [18] etc. The above methods are also common in practical application. In [7, 28], IMU data is collected through the wearable inertial sensor, and the information is extracted from continuous IMU samples by using fixed sliding window size, however, the results lack theoretical support and experimental proof. A method of identifying the sub-health status of subway doors by multi-scale sliding window algorithm is provided in [26], in this paper, only the initial window size was set, without considering how to determine the multi-scale, and the multi-scale problem played an important role in identification. The sliding window size and parameter setting in the prediction hydrological time series algorithm is discussed by Vafaeipour [25], the optimal window width is still derived from experience, and the application situation is limited, which is not universal. U. Yun and G. Lee etc mined the high average utility pattern [21, 30–32, 34] and weighted maximal frequent pattern [12, 33] based on the sliding window form the data streams. The algorithms considered the latest data stream information and weight conditions by employing a tree structure, the runtime, memory consume and scalability of the algorithms proposed by the authors were greatly improved. However, the approaches only focus on the fixed sliding window size but do not deal with optimal width also. Meanwhile, the proposed method can consider the bidirectional together with the weight conditions of items and sliding window-based streams.

All the above research results are based on the hypothesis that the initial size of the sliding window set by empirical value. Such strategy cannot reappear in practice, it is very likely that the information points with large deviation and valuable message cannot be effectively retained, and the influence of sliding window size on time series similarity calculation is often ignored. Therefore, it is necessary to find a method to determine the sliding window size in the existing time series dimension reduction methods in order to keep the data undistorted and reduce the computational complexity.

In this paper, the relevant definitions are firstly expounded, and then, on the basis of analyzing the deficiencies of existing methods, an algorithm is proposed to determine the initial size of the sliding window. To acquire the higher precision during the time series similarity division, the weight is also introduced to the LB-Hust distance algorithm. Finally, the verification is carried out in several aspects, and the experimental results are analyzed and summarized.

3. Related definitions.

Definition 3.1. Time series: an ordered set of elements arranged by timestamp in the sliding window, that is, time series [15], recorded as $S_n = (S_1 = (V_1, T_1)\,(S_2 = (V_2, T_2)\,, \ldots, (S_i = (V_i, T_i))$ among them $(S_i = (V_i, T_i))$represents the property data value of V_i corresponding to the time T_i.

Definition 3.2. Sliding window: the time series with length of s and n variables is expressed as $s\,(t) = (S_1\,(t)\,, S_2\,(t)\,, \ldots, S_n\,(t))\,, 1 \leq t \leq s$. where, the j^{th} sliding window with length w of the variable i can be expressed as $\mathrm{s}\,(i, j) = (\mathrm{x}\,(i, j)\,, \mathrm{x}\,(i, (j + w)), \ldots, x(i, (j + w - 1))\,)$ it can be concluded that the whole time series contains sub-sequences of $(s - w + 1)$ [16].

Definition 3.3. DTW: DTW is the distance in the time axis that allows time series to curve towards it [23]. Suppose two time series q and p of length m and n, respectively,$q\,[1 : m] = \{q_1, q_2, \ldots, q_m\}\,, p\,[1 : n] = \{p_1, p_2, \ldots, p_n\}$, the distance between the two can construct a matrix of size $m * n$ and $D\,(q, p)$ is the matching distance corresponding to the optimal matching path.

Definition 3.4. LB_Hust distance: the LB_Hust distance is improved based on the DTW distance and is a symmetrical lower bound distance [17]. Such as time series S_A and S_B, its lower bound distance has symmetry can be expressed as $D\,(S_A\,, S_B) = D\,(S_B\,, S_A)$.

Definition 3.5. Early stop strategy: he cumulative distance between two time series is generally increasing. Suppose that $S_A^{'}$ and $S_B^{'}$ are the subsequences of the time series S_A and S_B $(S_A^{'} < S_A; S_B^{'} < S_B)$ and calculated the LB_Hust distance between these subsequences, if $D_{LB\_Hust}(S_A^{'}, S_B^{'}) > \theta$ $(\theta\, is\, the\, threshold)$,the calculation can be stopped to reduce the CPU running time. On the contrary, if $D_{LB\_Hust}(S_A^{'}, S_B^{'}) \leq \theta$, the sequence is added to the set and the calculation is continued [2].

4. Algorithm research.

4.1. **General thinking.** Firstly, the boundary distance algorithm is adopted to determine the upper and lower boundaries of the normalized time series, to compress the processing range, and reduce the calculation amount of the subsequent similarity comparison. Then, the improved sliding window algorithm is designed to carry out the dimension reduction of the time series with determined boundary, to decide the alignment interval and improve the precision of segmentation. at last, the improved boundary distance algorithm is proposed to calculate the most similar matching subseries in the alignment interval, to ensure the precision of matching, provide reliable judgment basis for fault diagnosis, state evaluation, etc. the specific realization idea is as follows:

Step1 the time series S_A and S_B are normalized to obtain a better similarity comparison effect.

Step2 the LB_Hust algorithm was used to construct the upper and lower boundary curves U_1 and L_1, U_2 and L_2 for time series S_A and S_B.

Step3 the sliding window method is adopted to reduce the dimension of time series, the time series S_A and S_B with length N are traversed and divided into multiple sub-sequences by using the sliding window with size w, and the corresponding subsequences sets are $S_A^{'}, U_1^{'}, L_1^{'}, S_B^{'}, U_2^{'}, L_2^{'}$.

Step4 the LB_Hust algorithm is improved by weighting to calculate the processed time series and obtain the boundary distance $D_{LB\_Hust}(S_A^{'}, S_B^{'})$.

Step5 Suppose that θ is the empirical threshold, if $D_{LB\_Hust}(S_A^{'}, S_B^{'}) > \theta$,which means that the two series are not similar, and the calculation is stopped; if D_{LB_Hust} $(S_A^{'}, S_B^{'}) \leq \theta$,the subsequence is added to the set R as a similar sequence, and R is an n-dimensional array to load the subsequences.

The above process involves the improvement and optimization of boundary distance algorithm and sliding window size determination algorithm. The specific description is as follows:

4.2. **The boundary distance algorithm.** In order to find similar series, the upper and lower boundaries need to be determined, then the distance between the series waiting to be tested and the series as a sample is calculated according to the upper or lower boundary. Given two time series, the standard series, the standard series $S_A = \{a_1, a_2, \ldots, a_i, \ldots, a_n\}$, and the test sequence $S_B = \{b_1, b_2, \ldots, b_j, \ldots, b_m\}$, each component can be a number or a smaller vector. The curved path between series S_A and S_B is $W = \{w_1, w_2, \ldots, , w_k \ldots w_{\mathrm{K}}\}$, The k^{th} element is defined as $w_k = (i, j)_k$, $\max(m, n) \leq \mathrm{K} \leq m + n - 1$, and satisfied the following formula:

$$\mathrm{DTW}(S_A, S_B) = \min\left(\sqrt{\sum_{k=1}^{\mathrm{K}} w_k}\right) / \mathrm{K} \tag{1}$$

Formula 1 can be solved recursively, and the following analytical formula can be obtained

$$\begin{aligned} &P(a_1, b_1) = D(a_1, b_1) \\ &P(a_i, b_j) = D(a_i, b_j) + \min\left(P(a_{i-1}, b_j), P(a_{i-1}, b_{j-1}), P(a_i, b_{j-1})\right) \end{aligned} \tag{2}$$

The above algorithm named DTW adopted by Schultz D and Jain B [23], based on the idea of DTW, Kim [11] proposes one-dimensional query strategy to calculate the four feature vectors including maximum, minimum, starting value and end value, and the Euclidean distance index of corresponding features are constructed. Roh [20] calculated the distance between one time series data set and another with maximum and minimum. Mueen [8] improved the DTW algorithm by introducing the upper boundary U and lower boundary L to limit the value range of the curved path w. The upper and lower boundaries of the i-section are defined as follows:

$$U_i = \mathrm{MAX}(X_{i-w}, X_{i+w}) \tag{3}$$

$$L_i = \mathrm{MIN}(X_{i-w}, X_{i+w}) \tag{4}$$

The width of the interval is Ui minus Li, in different cases, the width is different. In a certain period, the amplitude difference is large, the width is wide, the amplitude is gentle, and the width is narrow. Two time seriesS_A and S_B of length n, among them $a_i \in S_A, b_j \in S_B$ and $j - w \leq i \leq j + w$, the lower bound distance $\mathrm{D}_{\mathrm{LB\_Keogh}}(S_A, S_B)$ is defined as follows:

$$LB\_Keogh(S_A, S_B) = \sqrt{\sum_{i=1}^{n} \begin{cases} (\mathrm{b}_i - U_i)^2 if \mathrm{b}_i < L_i \\ (\mathrm{b}_i - L_i)^2 if \mathrm{b}_i < L_i \\ 0 otherwise \end{cases}} \tag{5}$$

The analysis shows that LB_Keogh distance has no symmetry, that is D_{LB_Keogh} $(S_A, S_B) \neq D_{LB\_Keogh}(S_B, S_A)$, the asymmetry makes the calculation times and

complexity increase, which is not convenient for cluster analysis. The method of LB_Hust based onLB_Keogh distance algorithm meets the requirement of symmetry, and the time complexity of the LB_Hust method is $O(n)$, which is better than DTW method with the time complexity of $O(n*m)$. Two time series S_A and S_B of length n, among them $a_i \in S_A, b_j \in S_B$, and $j-w \le i \le j+w$, then its lower bound distance$\mathrm{D}_{\mathrm{LB_Hust}}(S_A, S_B)$, is defined as follows:

$$LB\_Hust(S_A, S_B) = \sqrt{\sum_{i=1}^{n} \begin{cases} (L_{a_i} - U_{b_i})^2 if L_{a_i} > U_{b_i} \\ (L_{b_i} - U_{a_i})^2 if L_{b_i} < U_{a_i} \\ 0 otherwise \end{cases}} \tag{6}$$

We can reach the conclusion that $\mathrm{D}_{\mathrm{LB\_Hust}}(S_A, S_B) \le \mathrm{D}_{\mathrm{LB\_Keogh}}(S_A, S_B)$, and $\mathrm{D}_{LB\_\mathrm{Hust}}(S_A, S_B) = \mathrm{D}_{\mathrm{LB\_Hust}}(S_B, S_A)$, that's symmetry.

4.3. **Weighted boundary distance algorithm.** LB_Hust also has some deficiencies, which is one of the key parts of this paper to improve. As shown in figure 1, it is assumed that the lower boundary of time series A is L_A in its border index $U-L$, the upper boundary of time series B and C is U_B and U_C respectively in its boundary index $U-L$. then, the distance between time series A and B is the sum of the distance between the region X and the region Z, can be represented as $D_{AB} = D_X + D_Z$, the distance between the time series A and C is the sum of the distance between the region X and the region Y, it is $D_{AC} = D_X + D_Y$, the distance between the time series B and C is the sum of the distances between the region Y and the region Z, namely $D_{BC} = D_Y + D_Z$. clearly, $D_X > D_Z > D_Y$, the region Z is larger than the region Y, which leads us to make the conclusion that the distance between the time series A and B is greater than the distance between the time series A and C. However, in terms of trend direction, the distance between time series A and B should be smaller than that between time series A and C.

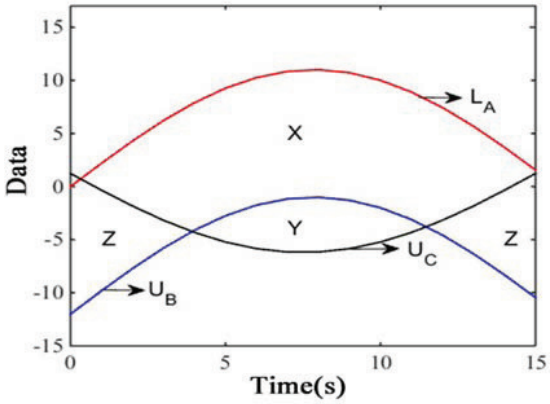

FIGURE 1. Distance between three series

After the above analysis, the original LB_Hust algorithm still has defects in practical application. In this paper, the trend is introduced into the original LB_Hust algorithm, and the distance between the two sequences is divided into a positive direction and a negative direction according to the trend. The distance between series was judged by both trend direction and numerical value. The formula of the

LB_Hust distance with trend is as follows:

$$D_{np} = w_n * D_n + w_p * D_p \tag{7}$$

w_n and w_p are the weight influence factors of the negative trend distance D_n and the positive trend distance D_p, and the weight coefficient satisfies: $w_n + w_p = 1$.In order to judge the plus and minus of the trend, the determination factor $Symbol$ is introduced, and the calculation formula is as follows:

$$Symbol = (\max U_{ai} - \min L_{ai}) * (\max U_{bi} - \min L_{bi}) \tag{8}$$

For the i^{th}element of a time series, $\max Ui$ and $\min Li$ reflect the trend of the curve in the time interval $[t_{i-w}, t_{i+w}]$. When $\max U_i > \min L_i$, the curve in the time interval $[t_{i-w}, t_{i+w}]$ shows an upward trend; when $\max U_i < \min L_i$, the curve in the time interval $[t_{i-w}, t_{i+w}]$ shows a downward trend.

The implementation process of the weighted LB_Hust algorithm with trend is as follows:

Algorithm 1 the weighted LB_Hust

input:
the negative trend distance, D_n
the positive trend distance, D_p
the weight influence factors, w_n,w_p
output:
the LB_Hust **distance with trend,**D_{np}

while $Len < length$ **do**
  if $U_{bi} > L_{ai}$ **then**
    Calculate the originalLB_Hust **distance:** $Dist = (U_{bi} - L_{ai})^2$
  else
    Calculate the originalLB_Hust **distance:** $Dist = (U_{ai} - L_{bi})^2$
  end if
  $Symbol = (\max U_{ai} - \min L_{ai}) * (\max U_{bi} - \min L_{bi})$
  if $Symbol < 0$ **then**
    $D_n += Dist$
  else
    $D_P += Dist$
  end if
  $len++$
end while
$D_{np} = w_n * D_n + w_p * D_p$

4.4. **Sliding window optimized algorithm.** When calculating the boundary distance and searching for similar series, reasonable segmentation of the time series will greatly reduce the amount of calculation. Although the sliding window algorithm is an effective method at present, the window size is set to a random value in existing methods, and the effect of the segmentation results on the calculation of boundary distance is not obvious. How to find the optimal sliding window size according to the actual demand is another focus of this paper.

The sliding window algorithm, by sliding L_s data values (L_s is steplength) backward through iteration in the series, forms $(s - w_s + 1)$ sub-sequences with length

w_s, continuously sliding $(s - w_s)/L_s$ times (s is the length of the series and w_s is the size of the window), forming an interval segment with equal step size and equal window size, and implementing time series segmentation. The principle is shown in figure 2.

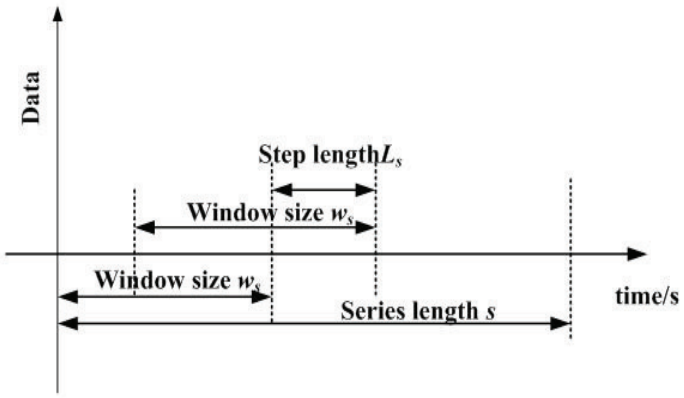

FIGURE 2. The sliding window principle

Generally, if time series S_A fluctuates little within a certain time range and the curve is relatively gentle, this interval can be judged as the segmentation point of the sliding window. In order to find the optimal window size, this paper assumes that the sliding window size $w_s = [2 : length(S_A)/2]$, and $steplength L_s = [0 : windowsize]$, the difference between the maximum value and the minimum value in a segment of a window is β. Then, if the difference in the window is less than the threshold sensibility, that is, $|S_{max} - S_{min}| \leq \beta$, then it can be the segmentation point, if the difference is greater than threshold, namely $|S_{max} - S_{min}| > \beta$, which cannot be the segmentation point, and add the difference into $DifferentArea$ collection. Then, determine whether there is a union between the $DifferentArea$ collection and the sliding window. If there is, illustrate the window size is not good, and stop the operation. If there is no union set, the current window size is added to the W set and the current $Steplength$ is added to the set of S. Finally, the value of window size and $Steplength$ left in the set are selected as the optimal values, and the experiments in the following part will verify the feasibility of the optimized algorithm proposed in this paper. The algorithm is described as follows:

5. **Experimental processes and analysis.** Experimental simulation environment: Win 7 Intel Core i3-7100 CPU @3.90GHz, Memory 4.00GB, Matlab 2017b.

5.1. **Experimental data.** To ensure the impartiality and reliability of the experiment, the selected data set was from the UCI school's named integrated time series dataset KDD archive [6], It consists of 600 instances of control points recorded by industrial process monitoring, which contain 6 different types of time series, they are normal, cyclic, increasing, decreasing, upward and downward, and there are 100 items in each type, and each item contains 60 data points. The properties of the time series mentioned above are shown in table 1, and the normalized sample state of 6 types time series is shown in figure 3.

Algorithm 2 Sliding Window Optimized Algorithm

Input: the parameter $Len\,(S_A)$
Output: the optimal value of $windowsize$ **and** $Steplength$

for $windowsize = 2Len\,(S_A)\,/2$ **do**
  for $Steplength = 0 : windowsize$ **do**
    $I = 1; J = windowsize$
    $I = 1 + Steplength$
    $J = 1 + windowsize + Steplength$
    $swArea = [I, J]$
    $DifferentArea = (S_{max} - S_{min} > \beta)$
    if $swArea \cap DifferentArea > 0$ **then**
      return
    end if
  end for
  if $swArea \cap DifferentArea \leq 0$ **then**
    $windowsize = windowsize + 1$
  end if
end for
$windowsize = max\,(windowsize)$
$Steplength = min\,(Steplength)$

TABLE 1. Dataset attribute

Type	Items	Status
1	1-100	Normal
2	101-200	Cyclic
3	201-300	Increasing trend
4	301-400	Decreasing trend
5	401-500	Upward shift
6	501-600	Downward shift

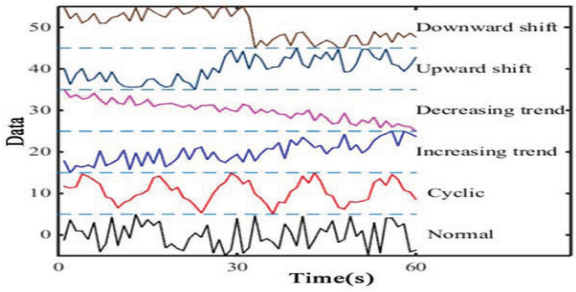

FIGURE 3. The normalized state of 6 types time serie

5.2. **Experimental results and analysis. Experiment 1:** Determine the Sliding Window Size and Steplength Range

The sliding window dimensions-reduction algorithm proposed above is used to calculate 6 types of synthetic control time series, in which the sliding window size $w_s = [2 : length\,(A)\,/2]$ and Steplength $L_s = [1 : w_s/2]$. As shown in FIG. 4,

the sliding window size is randomly generated by quadratic distribution, and the Steplength varies with the sliding window size, as shown in FIG. 5.

The specific steps are as follows: firstly, select a pair of randomly generated w_s and L_s within the specified range as the input of the algorithm, and store the 6 time series into the matrix T. Secondly, randomly select one time series in each category, calculate the distance between the six time series, store them in the matrix T_1, delete the six sequences that have been selected from the matrix T, and update the matrix T. Thirdly, repeat step 2 until the matrix T is empty and end loop. Fourthly, add up all distances in the matrix T_1, and divide by the number of times to obtain the average distance between classes D_{T1}. The calculation procedure of the average distance within the class is similar to that between the classes; we obtain the average distance within the classes D_{T2}. Then, the difference between D_{T1} and D_{T2} is $D_T = D_{T1} - D_{T2}$. The greater the D_T is the greater the average distance D_{T1} is and the smaller the average distance D_{T2} is, indicating that the selection of combination value is better. The corresponding distance D_T generated

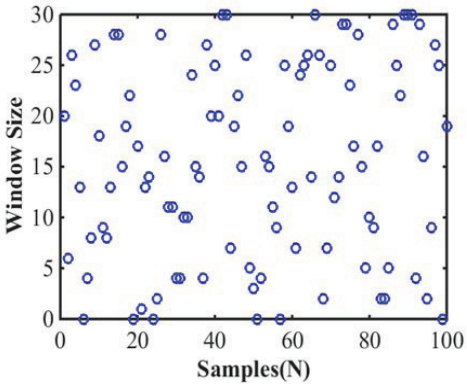

FIGURE 4. Quadratic distribution of window size

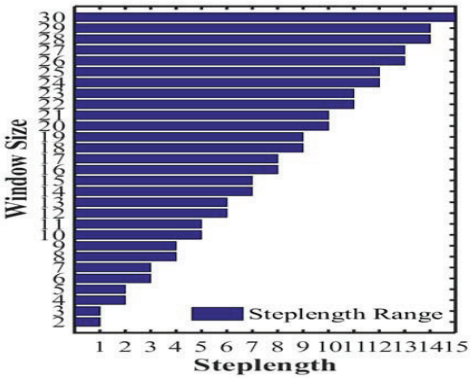

FIGURE 5. Steplength range

by the combination values of w_s and L_s is shown in table 2. In FIG. 6, the distance D_T obtained by combining all the values of w_s and L_s. It can be clearly seen from the figure that different combinations of values make the distance D_T different. We randomly selected the combination values of w_s and L_s and obtain the curve as depicted in FIG.7, where D_{T1}=94.7 and D_{T2}=29.1 when w_s =8 and L_s =2, $D_T = D_{T1} - D_{T2}$ =65.6 is the maximum value, that is, w_s =8 and L_s =2 is the optimal combination.

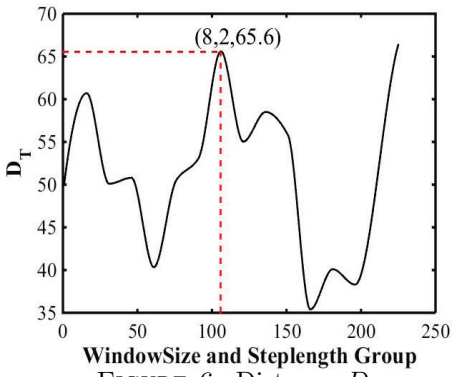

FIGURE 6. Distance D_T and values of w_s and L_s

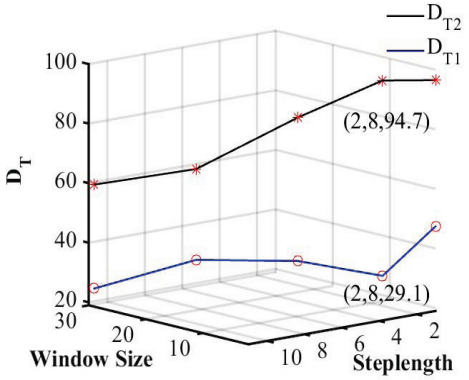

FIGURE 7. Cubic graph of Fig.6

TABLE 2. the combination value of w_s and L_s and the corresponding distance D_T

w_s \ L_s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	49.1	–	–	–	–	–	–	–	–	–	–	–	–	–	–
3	55.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–
4	54.8	60.6	–	–	–	–	–	–	–	–	–	–	–	–	–
5	58.7	61.2	–	–	–	–	–	–	–	–	–	–	–	–	–
6	64.3	65.3	60.0	–	–	–	–	–	–	–	–	–	–	–	–
7	62.5	63.9	61.3	–	–	–	–	–	–	–	–	–	–	–	–
8	64.8	65.6	64.0	62.6	–	–	–	–	–	–	–	–	–	–	–
9	62.5	61.2	63.5	62.3	–	–	–	–	–	–	–	–	–	–	–
10	63.4	63.9	59.1	62.9	60.8	–	–	–	–	–	–	–	–	–	–
11	58.0	61.9	62.1	58.5	60.1	–	–	–	–	–	–	–	–	–	–
12	61.3	61.2	58.3	61.1	60.9	59.1	–	–	–	–	–	–	–	–	–
13	60.2	59.9	59.1	49.3	58.2	49.3	–	–	–	–	–	–	–	–	–
14	49.9	59.6	60.2	60.8	59.1	60.1	61.7	–	–	–	–	–	–	–	–
15	55.9	56.2	53.2	48.2	60.2	49.2	58.7	–	–	–	–	–	–	–	–
16	60.9	55.2	58.3	58.0	52.1	50.0	56.1	49.0	–	–	–	–	–	–	–
17	49.2	53.1	54.4	55.2	60.8	59.4	51.7	53.3	–	–	–	–	–	–	–
18	53.7	54.4	52.0	52.3	52.8	60.1	59.2	49.2	49.2	–	–	–	–	–	–
19	58.3	60.8	55.1	50.3	58.7	58.0	58.1	53.0	51.9	–	–	–	–	–	–
20	50.7	53.3	55.8	50.1	42.3	45.7	50.0	51.2	51.2	46.3	–	–	–	–	–
21	49.6	50.2	46.9	46.5	47.2	47.1	47.5	46.0	48.2	43.9	–	–	–	–	–
22	51.3	49.9	48.2	46.7	50.0	45.0	40.0	39.2	39.5	40.7	39.1	–	–	–	–
23	39.9	54.2	50.0	49.8	49.0	36.8	36.5	38.1	42.5	43.9	35.9	–	–	–	–
24	46.8	51.9	48.3	45.0	38.2	42.7	39.4	50.2	41.9	38.4	43.3	51.8	–	–	–
25	51.6	40.0	58.7	43.1	40.0	39.4	35.0	45.3	45.9	41.2	38.1	40.9	–	–	–
26	47.3	48.6	50.3	39.6	42.6	55.2	42.0	36.1	35.0	42.0	43.8	39.5	47.8	–	–
27	47.5	51.3	40.0	41.6	39.5	35.0	50.0	49.2	39.4	38.4	35.6	39.2	49.9	–	–
28	45.0	40.4	38.4	35.0	35.7	46.2	50.6	45.2	39.1	39.6	42.1	48.2	40.0	38.9	–
29	50.1	48.3	40.2	41.6	35.9	36.1	40.3	39.4	50.1	46.3	39.6	35.9	35.9	35.0	–
30	42.2	49.8	45.0	39.2	40.0	38.9	40.4	39.3	37.5	38.6	36.3	36.9	35.0	36.2	35.2

Experiment 2: The LB_Hust Distance Calculation with Weight Coefficient.

After the analysis in section 4.3, we know that, besides the sliding window size, the weight coefficient plays an important role in LB_Hust algorithm when calculating the distance, The value of w_n determines the influence of the negative trend

distance D_n , and the value of $w_p$determines the influence of the positive trend distance D_p, when calculating the series similarity, both of them determine which component has a greater impact on the operation of the LB_Hust distance. The following experiment will discuss how to determine these two weight coefficients.

In order to evaluate the effectiveness of this method, the SCADA data set was taken as an example which generated during the operation of generator bearings of a wind turbine unit 5, 6, 7 in north China, the three series formed in a certain time period are shown in figure 8. The wind speed trend measured by no. 5 generator is different from that of no. 6 generator, but the wind speed measured by no.5 generator is less different from that of no.7 generator, with only phase difference. The original LB_Hust distance algorithm and LB_Hust distance algorithm with different weight coefficients were adopted to get the distance sets, and then k-means clustering analysis was conducted on them.

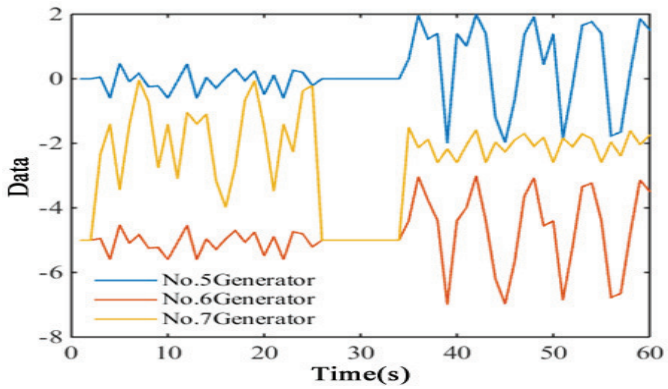

FIGURE 8. Time series of the three generators

The clustering result of the original LB_Hust distance is shown in figure 9(a), No.5 and No.6 belong to one class, and No.7 is another. When w_n=w_p=0.5, the clustering result of the weighted LB_Hust distance is shown in figure 9(b), the clustering results were similar to the original LB_Hust distance, in other words, the original LB_Hustt distance is equal to the weighted LB_Hust distance with w_n=w_p=0.5. When w_n=0.6 and w_p=0.4, the clustering result of the weighted LB_Hust distance is shown in figure 9(c), No.5 and No.7 belong to the same category, and No.6 is another, which is in line with the actual situation.

To further illustrate the range of these two weight coefficients, according to the KDD archive dataset of UCI with 6 types series adopted in experiment 1, the error rate of the weighted LB_Hust distance method was analyzed under the different value of w_n and w_p, the clustering error rate is shown in figure 10. When $w_n > w_p$, the weighted LB_Hust distance algorithm takes into account the difference in trend while calculating the numerical comparison, the error rate goes down, and when w_n = 0.6, w_p=0.4, the clustering error rate is lowest, with the continuous increase of w_n, the clustering error rate increased. Therefore, it can be concluded that under different data sets, the weight coefficient is the most reasonable when w_n=0.6 and w_p=0.4

Experiment 3: Weighted Boundary Distance Algorithm Based on Optimized Sliding Window Size.

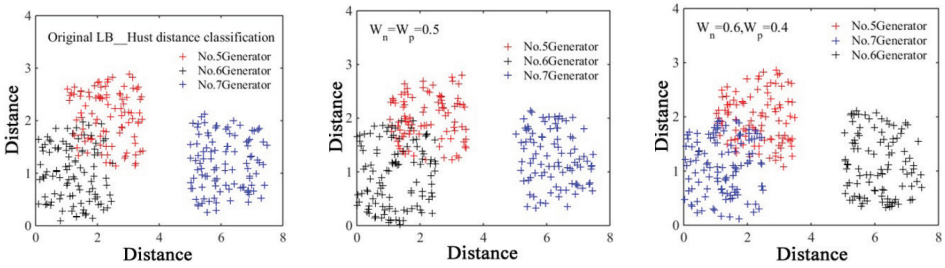

FIGURE 9. Clustering result comparison

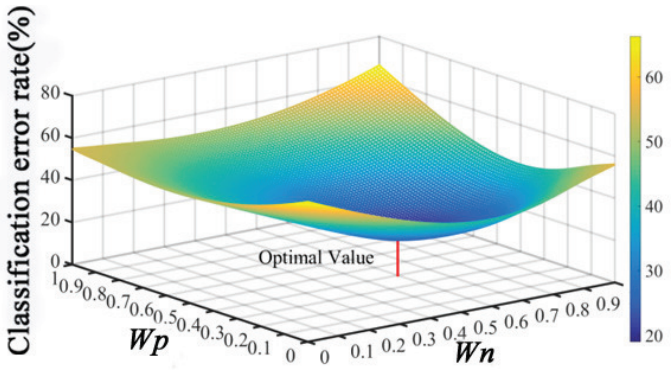

FIGURE 10. Clustering error rate with different weight coefficients

It is need to choose the three optimal parameters, they are sliding window size w_s negative trend w_n, positive trend w_p when using weighted LB_Hust algorithm for time series similarity measurement, this experiment adopts 5 groups of data sets, the attribute as shown in the table 3, the k-fold cross-validation (k-a fold cross-validation) method is adopted to select these parameters, generally take $k = 10$, the concrete steps are as follows:

Firstly, the data set Q is divided into k disjoint subsets. Assuming that the number of training samples in Q is m, and then there are m/k training samples for each subset. The corresponding subset is called $\{s_1, s_2, \ldots, s_k\}$. Each time, take one of the subsets as the test set, and the other $k1$ as the training set, and so on, until each of them is the test set.

Secondly, make the parameters values of w_s, w_n, and w_p within a certain range. Take the training set as the original data set and use the k-step cross-validation method to obtain the validation precision of the training set under multiple groups of w_s, w_n, and w_p. Select the group with the highest precision as the optimal parameter.

Thirdly, put this group of optimal parameter into the test set to obtain the precision, and calculate the average of similarity precision obtained k times as the true precision. $Mean\_precision = Sum\_precision/k$.

Table 4 shows the results of the 10-step cross-validation experiment. It can be seen that the average accuracy of each test set is lower than that of the training set.

When the w_s, w_n, and w_p parameters of the temperature, position, concentration and flow rate data sets are selected as 8,0.6 and 0.4, the precision is higher than that of the w_s, w_n, and w_p parameters of the pressure data set, which are selected as 9,0.6 and 0.4 respectively. FIG. 11 is an illustration of the precision obtained by combining all the values of w_s, w_n, and w_p. It can be seen more clearly from FIG. 11 that the precision varies with different combinations of values, and the precision is distributed between 0.1 and 1, where the highest precision is 0.92 when w_s=8, w_n=0.6 and w_p=0.4.

TABLE 3. 5 Groups Dataset

Dataset	Samples	Categories	Attributes
temperature	148	3	2
pressure	169	4	12
position	327	10	17
concentration	112	6	16
flow rate	236	5	7

TABLE 4. Cross-validation results

Dataset	The optimal value			Average precision	
	w_s	w_n	w_p	test set	training set
temperature	8	0.6	0.4	90%	92%
pressure	9	0.6	0.4	89%	91%
position	8	0.6	0.4	91%	92%
concentration	8	0.6	0.4	90%	91%
flow rate	8	0.6	0.4	90%	92%

FIG. 11 shows the different precision of the five distance calculation methods in the five data sets when the window size w_s=8, the positive and negative trend factor w_n=0.6 and w_p=0.4. Experimental results show that the weighted LB_Hust algorithm has a higher precision in all five data sets, especially in the temperature data set, according to the sliding window initial size determination algorithm, when the window size w_s =8, step length L_s=2, the positive and negative trend factor of the weighted LB_Hust algorithm is w_n =0.6 and w_p=0.4, we can get the precision of the algorithm on five data sets was 0.92, 0.87, 0.75, 0.80 and 0.70 respectively, thats the optimal value, which is consistent with experiment 1 and experiment 2.

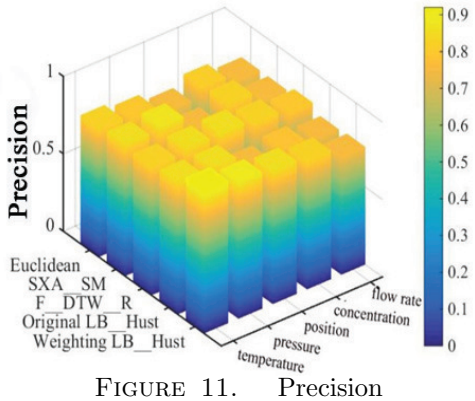

FIGURE 11. Precision of five methods on5 data sets

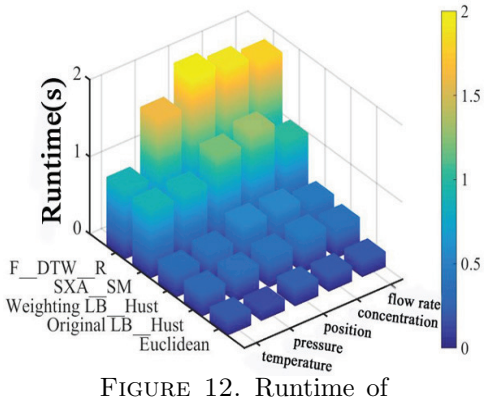

FIGURE 12. Runtime of five methods on 5 data sets

High time complexity is an urgent problem to be solved in the similarity measurement of time series, which concerns the feasibility when applied to large data sets. the algorithm LB_Hust optimize time complexity greatly, because the weighted LB_Hust sequence point matching relation is clear, don't need to be considered the effect of computing dynamic time warping and, the time complexity changed from $O(nm)$ (the traditional DTW distance) to $O(n)$ when find the shortest path, n and m respectively correspond to the length of two time series for comparison. In the application of high-dimensional time series comparison, the reduction of time complexity is very considerable. At the same time, as the sliding window performs a good dimensional-reduction processing on the sequence and the forming process of UL boundary index is also improved, the total time complexity of the weighted LB_Hust distance of the sliding window is $O(3n) + O(n)$, however, The time complexity of the original LB_Hust distance is $O(n * 2w_s) + O(n)$, and w_s is the size of the sliding window.

As is shown in figure 12, with different data set, the running time of the five methods is increasing, the running time of the original LB_Hust and weighted LB_Hust algorithms is comparable, as the complexity of the weighted distance algorithm increases, its running time will increase compared with the original distance algorithm, but it is still better and acceptable compared with other algorithms, The running time of the F_DTW_R algorithm [10] and the SXA_SM algorithm [9] is much longer than the weighted LB_Hust algorithm. Experimental results show that the proposed method is feasible.

6. **Conclusions.** Based on the traditional algorithm, a weighted boundary distance algorithm with determined sliding window size was proposed in this paper. This algorithm can determine the initial size of the sliding window by the statistical characteristics of time series, which can keep important data points effectively and divide time series reasonably. Whats more, on the basis of LB_Hust algorithm, the bidirectional weight coefficient is introduced according to series trend characteristics, which makes it more accurate in time series similarity calculation. Experimental results show that the weighted LB_Hust algorithm with determined window size has better indexes in terms of clustering precision and recall rate than the original LB_Hust algorithm, compared with other algorithms; the algorithm

also has a strong advantage in time complexity, and can effectively improve the efficiency of series similarity calculation. The next step is to introduce the method to the fault diagnosis of mechanical equipment to improve the precision and timeliness of detection.

Acknowledgments. This paper is supported by Natural Science Foundation of China (No. 61771492, No. 61871432), the Natural Science Foundation of Hunan Province (No.2017JJ3065, No.2019JJ6008, and No.2019JJ60054), 2018 China scholarship council higher education teaching method research project, and 2017 Zhuzhou science and technology project.

REFERENCES

[1] R. Belohlavek and V. Vychodil, Relational similarity-based model of data part 1: Foundations and query systems, *Int. J. General Syst.*, **46** (2017), 671–751.

[2] W. Bian and D. Tao, Max-Min distance analysis by using sequential sdp relaxation for dimension reduction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33** (2011), 1037–1050.

[3] M. R. Chernick, Wavelet Methods for Time Series Analysis, *Technometrics*, **43** (2016), 491–508.

[4] L. Dong, S. Liu and H. Zhang, A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples, *Pattern Recognition*, **64** (2017), 374–385.

[5] G. Hesamian and M. G. Akbari, A semi-parametric model for time series based on fuzzy data, *IEEE Transactions on Fuzzy Systems*, **99** (2018), 1–10.

[6] K. Hornik, I. Feinerer and M. Kober, et al., Spherical k-means clustering, *J. Statistical Software*, **50** (2017), 1–22.

[7] B. Hu, P. C. Dixon and J. V. Jacobs, et al., Machine learning algorithms based on signals from a single wearable inertial sensor can detect surface- and age-related differences in walking, *J. Biomechanics*, **71** (2018), 36–48.

[8] I. Güler and M. Meghdadi, A different approach to off-line handwritten signature verification using the optimal dynamic time warping algorithm, *Digital Signal Processing*, **18** (2008), 940–950.

[9] H. Ji, C. Zhou and Z. Liu, An approximate representation method for time series symbol aggregation based on the distance between origin and end, *Computer Science*, **10** (2018), 135–147.

[10] R. J. Kate, Using dynamic time warping distances as features for improved time series classification, *Data Min. Knowl. Discov.*, **30** (2016), 283–312.

[11] S. W. Kim, J. Kim and S. Park, Physical database design for efficient time-series similarity search, *IEICE Trans Commun.*, **91** (2008), 1251–1254.

[12] G. Lee, U. Yun and K. Ryu, Sliding window based weighted maximal frequent pattern mining over data streams, *Expert Syst. Appl.*, **41** (2014), 694–708.

[13] R. Li, X. Wu and S. Yang, Dynamic on-state resistance test and evaluation of GaN power devices under hard and soft switching conditions by double and multiple pulses, *IEEE Transactions on Power Electronics*, **34** (2018), 1–6.

[14] T. Luo, C. Hou and F. Nie, Dimension reduction for non-Gaussian data by adaptive discriminative analysis, *IEEE Transactions on Cybernetics*, **49** (2018), 1–14.

[15] M. D. C. Moura, E. Zio and I. D. Lins, et al., Failure and reliability prediction by support vector machines regression of time series data, *Reliability Engineering and Syst. Safety*, **96** (2017), 1527–1534.

[16] S. J. Noh, D. Shim and M. Jeon, Adaptive sliding-window strategy for vehicle detection in highway environments, *IEEE Transactions on Intelligent Transportation Syst.*, **17** (2016), 323–335.

[17] N. M. Parthaláin, Q. Shen and R. Jensen, A distance measure approach to exploring the rough set boundary region for attribute reduction, *IEEE Transactions on Knowledge and Data Engineering*, **22** (2010), 305–317.

[18] F. Petitjean, G. Forestier and G. I. Webb, Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm, *Knowledge and Information Syst.*, **47** (2016), 1–26.

[19] H. Ren, M. Liu and Z. Li, A piecewise aggregate pattern representation approach for anomaly detection in time series, *Knowledge-Based Syst.*, **21** (2017), 213–220.
[20] J. W. Roh and B. K. Yi, Efficient indexing of interval time sequences, *Inform. Process. Lett.*, **109** (2008), 1–12.
[21] H. Ryang and U. Yun, High utility pattern mining over data streams with sliding window technique, *Expert Syst. Appl.*, **57** (2016), 214–231.
[22] M. J. Safari, F. A. Davani and H. Afarideh, Discrete fourier transform method for discrimination of digital scintillation pulses in mixed neutron-gamma fields, *IEEE Transactions on Nuclear Science*, **63** (2016), 325–332.
[23] D. Schultz and B. Jain, Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces, *Pattern Recognition*, **74** (2018), 340–358.
[24] Y. Sun, J. Li and J. Liu, An improvement of symbolic aggregate approximation distance measure for time series, *Neurocomputing*, **102** (2014), 189–198.
[25] M. Vafaeipour, O. Rahbari and M. A. Rosen, et al., Application of sliding window technique for prediction of wind velocity time series, *Inter. J. Energy and Environmental Engineering*, **5** (2014), 105–116.
[26] Y. Xue, X. Mei and Y. Zhi, Method of subway health status recognition based on time series data mining, *Information Sciences*, **38** (2018), 905–910.
[27] G. Yuan, P. Sun and J. Zhao, A review of moving object trajectory clustering algorithms, *Artificial Intelligence Review*, **47** (2017), 1–22.
[28] R. Yao, G. Lin and Q. Shi, Efficient dense labelling of human activity sequences from wearables using fully convolutional networks, *Pattern Recognition*, **78** (2017), 221–232.
[29] S. Yue, Y. Li and Q. Yang, Comparative analysis of core loss calculation methods for magnetic materials under no sinusoidal excitations, *IEEE Transactions on Magnetics*, **54** (2018), 1–5.
[30] U. Yun, D. Kim, H. Ryang, G. Lee and K. Lee, Mining recent high average utility patterns based on sliding window from stream data, *J. Intelligent and Fuzzy Syst.*, **30** (2016), 3605–3617.
[31] U. Yun, D. Kim, E. Yoon and H. Fujita, Damped window based high average utility pattern mining over data streams, *Knowl.-Based Syst.*, **144** (2018), 188–205.
[32] U. Yun and G. Lee, Sliding window based weighted erasable stream pattern mining for stream data applications, *Future Generation Comp. Syst.*, **59** (2016), 1–20.
[33] U. Yun, G. Lee and K. Ryu, Mining maximal frequent patterns by considering weight conditions over data streams, *Knowl.-Based Syst.*, **55** (2014), 49–65.
[34] U. Yun, G. Lee and E. Yoon, Efficient high utility pattern mining for establishing manufacturing plans with sliding window control, *IEEE Trans. Industrial Electronics*, **64** (2017), 7239–7249.
[35] M. Zhu, D. G. M. Mitchell and M. Lentmaier, Braided convolutional codes with sliding window decoding, *IEEE Trans. on Communications*, **65** (2017), 3645–3658.

Received October 2018; 1st revision February 2019; 2nd revision April 2019.

E-mail address: chengpeng@csu.edu.cn
E-mail address: zhtang@csu.edu.cn
E-mail address: gwh@csu.edu.cn
E-mail address: 292702368@qq.com
E-mail address: 10822060@qq.com