Detecting Major Extrema in Streaming Time Series

Bui Cong Giao(✉) and Ho Van Cuu

Faculty of Electronics and Telecommunications, Saigon University,
Ho Chi Minh City, Vietnam
{bcgiao,cuuhovan}@sgu.edu.vn

Abstract. Time series are formed from data points collected over time. The prominent data points of time series are often minima or maxima; hence they have special values. Moreover, they are virtually turning points that change trend of time series. These prominent data points play an important role in determining the characteristics of time series so they are called important data points or major extrema. There are many methods to detect major extrema in time series in static context; however, in streaming context there have almost been no methods to carry out this task so far. In the paper, we propose a method for detecting major extrema in streaming time series. The method is of low computational time in identifying major extrema as soon as a newly in-coming data point of streaming time series is collected. The experimental results demonstrate that the proposed method exactly detects major extrema on the fly. Furthermore, the method could identify correlation of streaming time series thanks to their major extrema. An interesting application of the proposed method is to enable the task of online forecasting to predict future data points of streaming time series based on similarity search using major extrema.

Keywords: Major extrema · Streaming time series

1 Introduction

Problems of data mining on streaming time-series often appear in real-time processing applications which almost require fast and accurate response times. Solutions to these problems use frequently techniques to accelerate the computation of distances between time series [1,2], and appropriately segment streaming time series into subsequences [3] for further tasks of data mining on streaming time series. Therefore, the time-series segmentation is an important preprocessing step for such tasks, such as similarity search [4], anomaly detection [3], online forecasting [5], etc. The segmentation techniques employ frequently critical data points of time series, called major extrema, to improve the performance of these tasks.

supported by Saigon University.

C. V. Phan and T. D. Nguyen (Eds.): ICTCC 2022, LNICST 473, pp. 61–78, 2023.
https://doi.org/10.1007/978-3-031-28790-9_5

Major extrema of time series are often prominent, distinct from data points of average amplitude. Moreover, they are often local important minimum or maximum data points. As for static setting, there are a lot of research studies on how to detect major extrema in time series [6,7]; however, there have virtually been not methods to perform the task in streaming time series up to now. Motivated by the above observation, in the paper we propose an efficient method to detect major extrema in streaming time series.

The main contributions of the paper are as follows:

i. A proposed method to identify major extrema in streaming time series on the fly. The method could detect major extrema quickly when there is a newly incoming data point of streaming time series.
ii. The application of major extrema to find out the correlation between two streaming time series, and to segment streaming time series into subsequences for online forecasting.

The paper is structured as follows. Section 2 reviews related works. Section 3 briefly describes some basic background. In Sect. 4 we present the proposed method. Section 5 reports the experimental evaluation. Finally, Sect. 6 gives conclusions and future work.

2 Related Works

There have been several typical research studies on major extrema and the application of major extrema for time-series data mining on both static and streaming background. These research studies would be reviewed in chronological order as below.

Fu et al. [6] claimed that data points with perceptually importance in the human visual identification process are more important than other data points in time series. The authors used perceptually important points (PIP) to build a framework representing financial time series. The framework helps to reduce the time-series dimension to different levels of detail based on PIP. PIPs of a time series are identified as follows. The first two PIPs are the starting point and the ending one of the time series. The successive PIP is the point with the greatest distance to the first two PIPs. The fourth PIP is the point with the greatest distance to its two adjacent PIPs. The process to find out remaining PIPs continues until all the points in the time series are visited or the required number of PIPs is reached. It is obvious that the method is solely suitable for static time series since in streaming scene, streaming time series are assumed that their data points come continuously and there is not the ending one.

Fink and Gandhi [7] presented the definitions of major extrema of time series and a one-pass algorithm to find major extrema. The algorithm is of low time complexity, $\mathcal{O}(n)$. Since the algorithm solely works on static background, it needs changing to identify major extrema of streaming time series.

Giao and Anh [5] proposed an online forecasting method in streaming time series based on similarity search [2]. The method takes every newly incoming

time-series subsequence of a streaming time series, then finds k nearest neighbor subsequences. Future data points of the streaming time series are forecast based on the k best matches. Before the similarity search, these subsequences have been retrieved from the streaming time series by a segmentation technique using major extrema. Note that in the research study the authors have not yet presented the method of identifying major extrema in streaming time series.

Zhan et al. [4] introduced an online segmenting algorithm for streaming time series. The algorithm segments streaming time series by choosing the most important turning point, which enable to reflect the variation trend features of the streaming time series. The segmenting approach is an improvement of online Piecewise Linear Representation [8].

Thuy et al. [3] presented two methods, TopK-EP-ALeader and TopK-EP-ALeader-S, which combine segmentation and clustering for detecting top-k discords in static and streaming time series. The segmentation is based on the major extrema method of Fink and Gandhi [7]. However, like the article [5], the article [3] also do not present the technique of identifying major extrema in streaming time series.

3 Background

The section presents some definitions of streaming time series, major extrema, and online forecasting using major extrema to segment streaming time series into subsequences.

3.1 Streaming Time Series

Definition 1. *(Streaming time series)* A streaming time series X is a discrete, semi-infinite numerical sequence obtained from collecting or sampling a data stream at specific time ticks. Mathematically, X is represented by data points whose values are real numbers: $x_1, x_2, \ldots, x_n \ldots$, where x_n is the most recent data point, called the newly incoming one.

A subsequence C of X is a time series in X, i.e. $C = x_i, x_{i+1}, \ldots, x_j$ for $1 \leq i$ and $j \leq n$. For simplicity, let $(x_i : x_j)$ denote $\{x_i, x_{i+1}, \ldots, x_j\}$. It is noted that tasks of data mining on streaming time series often consider the newly incoming subsequence $(x_i : x_n)$.

3.2 Major Extrema

Extrema of a time series are data points of local minimum or maximum. According to Fink and Gandhi [7], *strict*, *left*, *right* and *flat* minima are defined as below.

Definition 2. *(Minima)* Given time series $X = (x_1 : x_n)$ and $1 < i < n$.

- x_i is strict minimum if $x_{i-1} > x_i < x_{i+1}$.

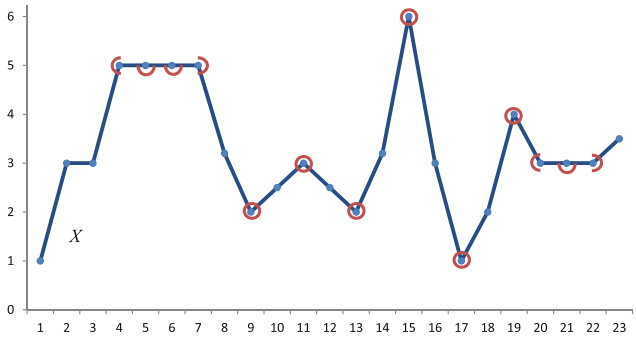

Fig. 1. An illustration of time series X and its extrema

- x_i is left minimum if $x_{i-1} > x_i$ and there is r, where $r > i$, such that $x_i = x_{i+1} = \cdots = x_r < x_{r+1}$.
- x_i is right minimum if $x_i < x_{i+1}$ and there is l, where $l < i$, such that $x_{l-1} > x_l = \cdots = x_{i-1} = x_i$.
- x_i is flat minimum if there are l and r, where $l < i < r$, such that $x_{l-1} > x_l = \cdots = x_{i-1} = x_i = x_{i+1} = \cdots = x_r < x_{r+1}$.

The definition of maxima is similar to that of minima. Figure 1 depicts time series X and its extrema as follows. Strict extrema are shown as circles, left and right as half-circles and flat extrema as downward half-circles.

A time series might have a lot of extrema; however, major extrema are often more interested. Fink and Gandhi [7] defined major extrema as below.

Definition 3. *(Important minima)* Given a distance function $d(a, b) = |a - b|$ and a positive value R, x_i is an important minimum if there are l and r, where $l < i < r$, such that x_i is a minimum among $x_l, \ldots, x_r$ and $d(x_i, x_l) \geq R$ and $d(x_i, x_r) \geq R$.

There are some important notes about Definition 3:

- The parameter R is a *compression rate* to determine important minima. The larger R is, the fewer the number of important minima is identified. To illustrate the compression rate, Fig. 2 depicts examples of the four types of important minima that are identified by R. Because major extrema are often prominent, different from data points of average amplitude, it is reasonable to set

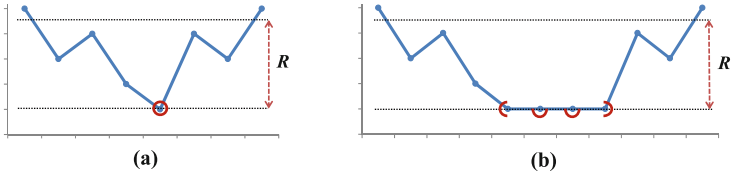

Fig. 2. (a) Strict important minimum (b) Left, flat, and right important minima [5]

$R = \beta \times \sigma$, where β is the *tuning* parameter and often larger than 1, and σ is the standard deviation of the time series. If a time series does not change much, β might be smaller than 1 to get more major extrema.

- Flat important minima can be derived from left important minima and right important minima in case they are consecutive as in Fig. 2 (b).
- The definition of important maxima can be inferred from Definition 3. Important minima and maxima are often called major extrema. Using major extrema of time series, we can segment time series sensibly and visually into subsequences

Given streaming time series X and a newly incoming data point x_n of X, the problem of our research study is how to fast detect major extrema right after the last major extremum x_k, where $k \leq n$.

The newly found major extrema often imply a certain special characteristic of X and they can be used to segment X for further data-mining tasks. The following online forecasting method [5] employs major extrema of X to segment a streaming time series into subsequences for the task of prediction.

3.3 Online Forecasting

Given streaming time series X and a newly incoming data point x_n of X, the aim of online forecasting is to predict x_{n+p} for $p \geq 1$.

The online forecasting method [5] has three main components: (i) k-NN or Simple Exponential Smoothing (SES), (ii) identifying major extrema, and (iii) similarity search under Dynamic Time Warping (DTW) [9].

To describe the method concisely, we need to review popular measures for evaluating accuracy of prediction, SES, and then introduce two definitions.

There are three common measures to evaluate accuracy of prediction: Mean Absolute Percentage Error (MAPE), Mean Absolute Deviation (MAD), and Mean Squared Error (MSE). For these measures, the smaller the value, the prediction method is more accurate. The meanings of the measures as follows. MAPE measures the accuracy of fitted time-series values. The accuracy is expressed as a percentage. MAD measures the accuracy of the prediction by averaging the alleged error. MSE measures the expected squared distance between the predicted value and the true one.

SES is a time-series forecasting method for univariate data without a trend or seasonality. The method is expressed by the following equation:

$$\hat{x}_{n+1} = \alpha x_n + (1 - \alpha)\hat{x}_n \tag{1}$$

where x_n is the most recent observation for time tick n, $\hat{x}_{n+1}$ is the smoothed value or the predicted value at time tick $n+1$. α is a smoothing constant whose value domain is from 0 and 1.

From Eq. 1, the next equation is

$$\hat{x}_{n+1} = \alpha x_n + \alpha(1-\alpha)x_{n-1} + \alpha(1-\alpha)^2 x_{n-2} + \cdots + \alpha(1-\alpha)^{n-1} x_1 + (1-\alpha)^n x_0. \quad (2)$$

To use Eqs. 1 and 2, we need to determine α and x_0 beforehand. The minimum results of the triple (MAPE, MAD, MSE) can be used to determine α. For an initial estimate of x_0, let x_0 be x_1, or be an average of a few previous observations.

The following are three necessary definitions for the online forecasting.

Definition 4. *(Segment)* A *segment* of a time series is a subsequence determined by two successive important maxima or two successive important minima.

Definition 5. *(Target subsequence)* A *target* subsequence is a newly incoming subsequence whose left end is the latest segment.

Definition 6. *(Source subsequence)* Given a *target* subsequence C, a *source* subsequence S for C have to satisfy three conditions:

(i) The data points of S must be in a buffer containing collected data points of X,
(ii) The left end of S is a major extremum of the same type as the major extremum at the left end of C, and
(iii) The two subsequences are the same length.

The working setting of the online forecasting method [5] is illustrated in Fig. 3. In addition, the method consists of two phases:

- *Sampling Phase:* In the beginning, all available data points of X are loaded into the buffer. Subsequently, the standard deviation σ is computed from these data points. Given a specific tuning parameter β, the compression rate R is then calculated from σ and β. Next, major extrema and segments are determined from the available data points of X. Finally, the target subsequence and its source ones are determined and normalized using z-score.
- *Forecasting Phase:* There are two cases are
  i. The newly incoming data point x_n of X does not cause any new major extrema. In the case, the target subsequence and its source ones are extended one data point to the right. After that, these subsequences are incrementally normalized [2].
  ii. The arrival of the data point x_n incurs new major extrema. In the case, the target subsequence needs redefining using the latest segment of X combined with x_n. Next, the source subsequences for the new target subsequence are identified, and then all of them are normalized from the scratch.

Having the target subsequence and its source ones, the similarity search method SUCR-DTW [2] conducts k-NN search to find k nearest neighbor subsequences of the target subsequence from the set containing the source ones. Subsequently, x_{n+p} is computed from data points following these k subsequences as

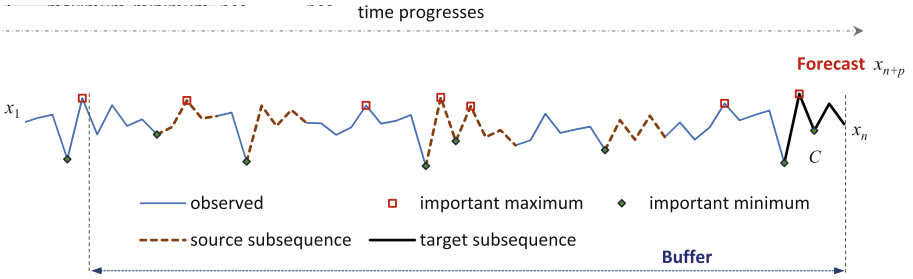

Fig. 3. The context for the operation of the online forecasting [5]

the following manner. Let S_i be one of the k nearest neighbor subsequences, and x_{si+p} be the data point following S_i by a distance of p time ticks. x_{si+p} is then normalized using the z-score coefficients of S_i. Let y_{si+p} be the normalized value of x_{si+p}. The normalized value y_{n+p} of x_{n+p} might be calculated in accordance with two ways:

The first way is average of the k normalized values.

$$y_{n+p} = \frac{1}{k} \sum_{i=1}^{k} y_{si+p}. \tag{3}$$

The second way is weighted average of the k normalized values.

$$y_{n+p} = \frac{2}{k \times (k+1)} \sum_{i=1}^{k} i \times y_{si+p}. \tag{4}$$

Having y_{n+p}, the data point x_{n+p} can be computed from the z-score coefficients of C. As for Eq. 4, it is worth noting that the closer to the target subsequence a resulting subsequence is, the greater the weight of the resulting subsequence is. Therefore, the online forecasting method has two variants, kNN-Av for Eq. 3 and kNN-WAv for Eq. 4. Furthermore, this method might hybridize with SES as follows.

$$y_{n+p}^{Hybrid} = \omega y_{n+p} + (1 - \omega) y_{n+p}^{SES} \tag{5}$$

where y_{n+p} is the normalized value by kNN-Av or kNN-WAv, y_{n+p}^{SES} is the normalized value by SES, and ω is a weighted parameter whose value is between 0 and 1.

The next section will reveal the proposed method to solve the problem of detecting major extrema in streaming time series on the fly.

4 Proposed Method

Let S be the ordered list of the major extrema of streaming time series X. The list follows an ascending index order of data points being major extrema. As mentioned earlier, flat important minima can be derived from the couples of

left important minima and right one that are consecutive as in Fig. 2 (b), so the proposed method uses only strict, left, and right minima and ignores flat ones. Similarity, flat important maxima could be ignored. Each element of S is (*i, detected time, type, extremum*), where i is the index of major extremum x_i, *detected time* is the time tick at which x_i are detected, *type* is strict or left or right, and *extremum* is minimum or maximum. Let R denote the compression rate. The appearance of newly incoming data point x_n triggers the execution of `Procedure Find_new_extrema`.

`Procedure Find_new_extrema`

---

1. if $S = \varnothing$ then
2. Find_first
3. if $S \neq \varnothing$ then Find_minmax
4. else
5. Find_minmax
6. end if

`Procedure Find_first` `// Find the first major extreme point`

---

1. $lMin = rMin = lMax = rMax = i = 1$
2. while $i < n$ and $d(x_{i+1}, x_{lMax}) < R$ and $d(x_{i+1}, x_{lMin}) < R$ do
3. i++
4. if $x_{lMin} > x_i$ then $lMin \leftarrow i$
5. if $x_{rMin} \geq x_i$ then $rMin \leftarrow i$
6. if $x_{lMax} < x_i$ then $lMax \leftarrow i$
7. if $x_{rMax} \leq x_i$ then $rMax \leftarrow i$
8. end while
9. i++
10. if $i < n$ and $x_i > x_1$ then Output_ext($lMin, rMin$, minimum)
11. if $i < n$ and $x_i < x_1$ then Output_ext$lMax, rMax$, maximum)

`Procedure Find_minmax` `// Find the next major extreme point`
`// right after the newly found major one:` x_k

```
1.  x_k ← the last item of S
2.  i ← k + 1
3.  if x_k is maximum then
4.     while i < n do
5.        i ← Find_min(i)
6.        if i < n then i ← Find_max(i)
7.     end while
8.  else
9.     while i < n do
10.       i ← Find_max(i)
11.       if i < n then i ← Find_min(i)
12.    end while
13. end if
```

```
Function Find_min(i)        // Find the next important minimum from
                            // x_i to x_{n-1}
```

```
1.  l = r = i
2.  while i < n and ( x_{i+1} < x_l or d(x_{i+1}, x_l) < R) do
3.     i++
4.     if x_l > x_i then l ← i
5.     if x_r ≥ x_i then r ← i
6.  end while
7.  if i < n then Output_ext(l, r, minimum)
8.  return i + 1
```

```
Function Find_max(i)        // Find the next important maximum from
                            // x_i to x_{n-1}
```

```
1.  l = r = i
2.  while i < n and ( x_{i+1} > x_l or d(x_{i+1}, x_l) < R) do
3.     i++
4.     if x_l < x_i then l ← i
5.     if x_r ≤ x_i then r ← i
```

```
6.  end while
7.  if i < n then Output_ext(l, r, maximum)
8.  return i + 1
```

```
Procedure Output_ext(l, r, extremum)      // Append the found major
                                          // extrema to S
```

```
1.  if l = r then
      // x_l is a major strict extremum
2.    Append (l, n, strict, extremum) to S
3.  else
      // x_l is a major left extremum
4.    Append (l, n, left, extremum) to S
      // x_r is a major right extremum
5.    Append (r, n, right, extremum) to S
6.  end if
```

`Procedure Find_minmax` shows that when a newly incoming x_n exists, the course of search for major extrema is conducted. Most frequently the search begins from x_{k+1}, where x_k is the newly found major extremum or the last item of S, to x_{n-1}. The time tick at which the major extrema are detected, that is n, is also recorded in `Procedure Output_ext`. The search is performed with the comparison operators (see lines 4 and 5 in `Functions Find_min` and `Find_max`) so the time complexity of `Procedure Find_new_extrema` is $\mathcal{O}(n-k) \approx \mathcal{O}(n)$. As for the memory used in the proposed method, it takes two buffers to contains data points. The first buffer stores data points of X. The buffer can be organized as a circular fashion and its size should be enough large to consists of data points which are still valuable. This means that the old data points of X that are not valuable would be deleted to make room for incoming data points. The second buffer is S consisting of major extrema. The buffer also works in a circular fashion as the first.

Note that the proposed method to identify major extrema in streaming time series can be used with the compression rate R as a constant or as a changeable value since some newly incoming data point. After a newly major extreme point x_i is found with a specific value of R, the successive major extreme point can be identified from the index $i + 1$ to the largest index n of X by another value of R. Besides, time-series data mining tasks using the proposed method need a sampling phase over the initial stage to emit data points of X so that σ can be computed on the corresponding initial subsequence of X. The size of the sampled subsequence is determined for each specific case study.

5 Experimental Evaluation

The section presents experiments on the proposed method with four streaming time series. These streaming time series are stimulated from static time series for working in streaming background. That is, there is a newly incoming data point collected from time series every time tick and the process of detecting new major extrema is performed immediately. Theses static time series may be downloaded from [10]. The experiments are divided into two kinds. The first illustrates results obtained from detecting major extrema in streaming time series of hydrology. The second shows an application of finding major extrema in streaming time series for prediction.

5.1 Detecting Major Extrema in Streaming Time Series of Hydrology

The two first experiments use time series of hydrology. They are *runoff_TriAn* and *runoff_PhuocHoa*. Theses time series are monthly runoffs from January 1978 to December 1993 collected by the two gauging-stations: Tri An and Phuoc Hoa located in the South of Vietnam. The data point 1 of *runoff_TriAn* is the runoff in January 1978 of Tri An. Likewise, the data point 2 of *runoff_TriAn* is the runoff in February 1978 of the gauging-station, etc. The time series thus has 192 data points. Similarity, *runoff_PhuocHoa* has 192 data points. The compression rate R is $\beta \times \sigma$, where σ is the standard deviation of time series, and β is the tuning parameter. In the two experiments, β is 1.1. Since the two time series are relatively short, we sampled entire data of them and got $\sigma = 236.684$ for *runoff_TriAn*, and $\sigma = 533.405$ for *runoff_PhuocHoa*.

Figure 4 shows the streaming time series presenting the monthly runoffs collected by Tri An and its major extrema. The important minima often happen in

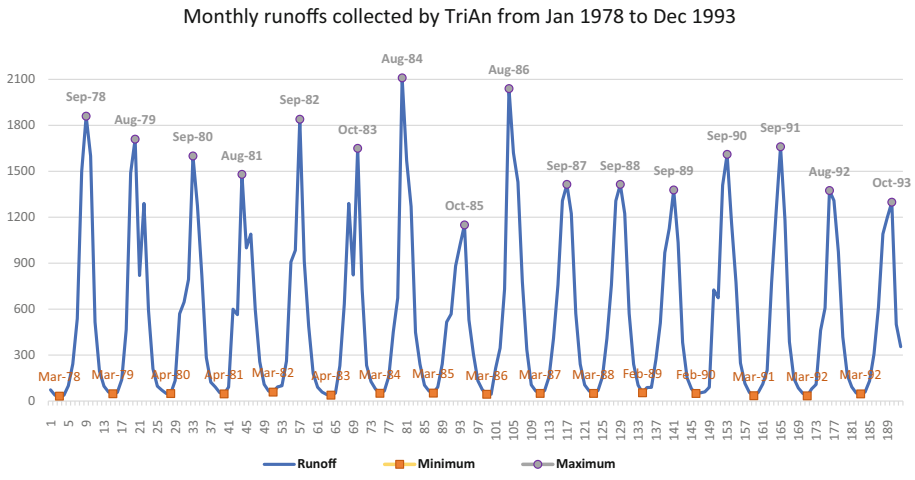

Fig. 4. Visualization of streaming time series *runoff_TriAn* and its major extrema

March and the important maxima often occur in August or September. Tri An records one minimal runoff and one maximal one for each year. Tri An gauges the highest runoff in August 1984.

Table 1 shows statistical results of the proposed method in streaming time series *runoff_TriAn*. The table identifies that the streaming time series has 16 important minima and 16 important maxima. Column Data point presents the index of the data points that are major extrema. Column Month-Year indicates the months corresponding to the data points being major extrema. Column Time tick of Detection depicts the time ticks at which the major extrema are discovered. For instance, the data point 3 corresponding to Mar-78 is an important

Table 1. Statistics of detecting major extrema in streaming time series *runoff_TriAn*.

No	Data point	Month-Year	Time tick of Detection	Runoff	Extremum
1	3	Mar-78	10	32	minimum
2	9	Sep-78	11	1,860	maximum
3	**15**	Mar-79	19	47.6	minimum
4	20	Aug-79	21	1,710	maximum
5	28	Apr-80	**31**	48.9	minimum
6	33	Sep-80	35	1,600	maximum
7	40	Apr-81	**44**	46.7	minimum
8	44	Aug-81	**47**	1480	maximum
9	51	Mar-82	55	58.6	minimum
10	57	Sep-82	58	1,840	maximum
11	64	Apr-83	**68**	38.5	minimum
12	70	Oct-83	**71**	1,650	maximum
13	75	Mar-84	**79**	50.7	minimum
14	80	Aug-84	**82**	2,110	maximum
15	87	Mar-85	92	52.1	minimum
16	94	Oct-85	**95**	1,150	maximum
17	**99**	Mar-86	**103**	44.4	minimum
18	104	Aug-86	106	2,040	maximum
19	111	Mar-87	115	49.91	minimum
20	117	Sep-87	119	1,414.59	maximum
21	123	Mar-88	**127**	54.16	minimum
22	**129**	Sep-88	131	1,378.21	maximum
23	**134**	Feb-89	139	22.82	minimum
24	141	Sep-89	143	710.37	maximum
25	**146**	Feb-90	150	49.49	minimum
26	153	Sep-90	**155**	1,610.66	maximum
27	**159**	Mar-91	163	34.9	minimum
28	165	Sep-91	167	1660.37	maximum
29	171	Mar-92	176	33.98	minimum
30	176	Aug-92	179	1,374.48	maximum
31	**183**	Mar-93	**188**	46.71	minimum
32	190	Oct-93	191	1,298.28	maximum

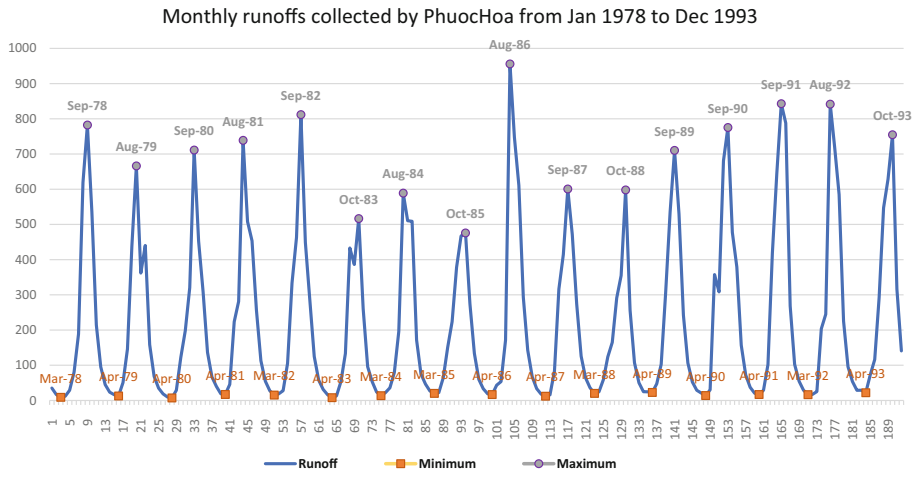

Fig. 5. Visualization of streaming time series *runoff_PhuocHoa* and its major extrema

minimum. The major extremum is detected at time tick 10. Column Runoff shows the values of major extrema.

Figure 5 represents the streaming time series indicating the monthly runoffs collected by Phuoc Hoa and its major extrema. The important minima often take place in March or April, and the important maxima appear frequently from August to October. Like Tri An, each year Phuoc Hoa has one important minima and one important maximal. Phuoc Hoa gauges the highest runoff in August 1986.

Table 2 illustrates statistical results of the proposed method in streaming time series *runoff_PhuocHoa*. The table also identifies that the streaming time series has 16 important minima and 16 important maxima. Note that the time ticks that appear major extrema of the two streaming time series are nearly the same. There are solely 8 times in which the major extrema of the two streaming time series take place in different months; however, the differences are negligible. For example, the 3rd major extremum of *runoff_TriAn* occurs in March 1979, whereas the 3rd major extremum of *runoff_PhuocHoa* takes place in April 1979. In addition, the time ticks at which the proposed method detects major extrema of the two streaming time series are nearly the same. There are only 12 out of 32 cases in which this method detects major extrema of the two streaming time series are different. For instance, the 5th major extremum of *runoff_TriAn* are identified at time tick 31 whereas the 5th major extremum of *runoff_PhuocHoa* are identified at time tick 32. It is obvious that the difference of the two detection time ticks is small. These differences are emphasized by the bold fonts in the two tables.

Table 2. Statistics of detecting major extrema in streaming time series *runoff_PhuocHoa*.

No	Data point	Month-Year	Time tick of Detection	Runoff	Extremum
1	3	Mar-78	10	8.89	minimum
2	9	Sep-78	11	782.33	maximum
3	**16**	Apr-79	19	12.76	minimum
4	20	Aug-79	21	666.19	maximum
5	28	Apr-80	**32**	7.18	minimum
6	33	Sep-80	35	711.27	maximum
7	40	Apr-81	**43**	17.04	minimum
8	44	Aug-81	**46**	738.87	maximum
9	51	Mar-82	55	15.24	minimum
10	57	Sep-82	58	811.87	maximum
11	64	Apr-83	**68**	8.01	minimum
12	70	Oct-83	**72**	516.66	maximum
13	75	Mar-84	**80**	13.58	minimum
14	80	Aug-84	**83**	588.94	maximum
15	87	Mar-85	92	20.15	minimum
16	94	Oct-85	**96**	475.84	maximum
17	**100**	Apr-86	**104**	16.8	minimum
18	104	Aug-86	106	955.9	maximum
19	**112**	Apr-87	115	12.03	minimum
20	117	Sep-87	119	600.63	maximum
21	123	Mar-88	**128**	20.05	minimum
22	**130**	Oct-88	131	598.06	maximum
23	**136**	Apr-89	139	22.82	minimum
24	141	Sep-89	143	710.37	maximum
25	**148**	Apr-90	150	13.94	minimum
26	153	Sep-90	**154**	775.27	maximum
27	**160**	Apr-91	163	16.82	minimum
28	165	Sep-91	167	843	maximum
29	171	Mar-92	176	16.45	minimum
30	176	Aug-92	179	841.68	maximum
31	**184**	Apr-93	**187**	22.02	minimum
32	190	Oct-93	191	754.77	maximum

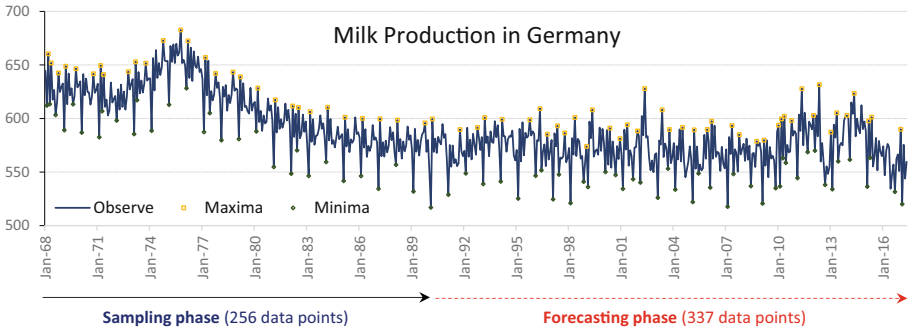

Fig. 6. Streaming time series *Milk Production in the Germany* and its major extrema

The appearance periods of the major extrema of the two streaming time series demonstrate that they have nearly the same trend. For this reason, these streaming time series have a strong correlation.

5.2 Online Forecasting in Streaming Time Series Using Major Extrema

The following are two experiments of the online forecasting method [5] to demonstrate an application of major extrema, which are detected on the fly, in segmenting streaming time series into subsequences. Note that only *single-step-ahead* prediction is considered in the experimental evaluation, that mean the online forecasting method tries to predict x_{n+1} where x_n is a newly incoming data point of streaming time series X.

Milk Production in the Germany. The dataset from the web page [11] is a time series of raw cows' milk productions in the Germany from January 1968 to May 2017. The time series thus has 593 data points. In the experiment, the size of the buffer is 256 and the tuning parameter β is 1.3. The standard deviation σ computed in the sampling phase is 29.733 so the compression rate R is 39.654. Figure 6 shows the monthly observations of the time series, the number of data points of the two phases, and the two working phases. The forecasting phase

Table 3. The best test cases of each method

Method	MAPE	MAD	MSE
kNN-Av with $k = 4$	2.183	12.455	286.928
kNN-WAv with $k = 5$	2.135	12.162	274.506
SES with $\alpha = 0.1$	2.667	15.154	383.638
Hybrid with $\omega = 0.9$	**2.130**	**12.138**	**268.198**

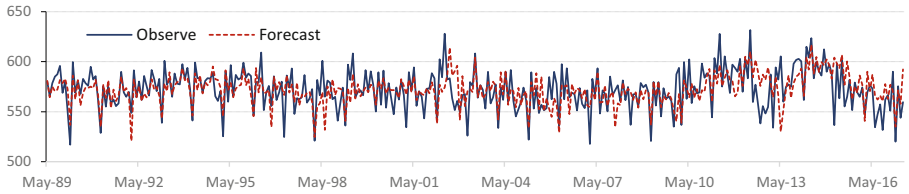

Fig. 7. The observations and predictions for *Milk Production in the Germany*

begins from May 1989 and spans 337 months. The number of important minima and maxima are 72, and 71, respectively.

It can be seen from Table 3, the hybrid method with $\omega = 0.9$ is the best test case. Notice that the hybrid method combines kNN-WAv with $k = 5$ and SES with $\alpha = 0.1$. The predictions obtained in the best test case of the hybrid method are illustrated in Fig. 7. The figure shows that the prediction time series and the observation time series have nearly the same fluctuation; however, their major extrema seldom coincide.

Temperatures at Savannah International Airport. The dataset from the web page [12] is a time series of monthly temperatures from January 1874 to May 2011. Therefore, the time series has 1,649 data points. In the experiment, the size of the buffer is 1,024, β is 1.3 and σ computed in the sampling phase is 6.327. Hence R is 8.225. Figure 8 shows the monthly observations of the time series and the number of data points of the two phases. The figure also shows the major extrema in the time series. The number of important minima and maxima are 138, and 144, respectively. The forecasts have been carried out since May 1959.

Table 4 shows the best prediction quality for each method. Note that although the hybrid method combines kNN-WAv with $k = 5$ and SES with $\alpha = 0.5$, the hybrid method with $\omega = 0.9$ does not give better results. kNN-WAv with $k = 5$ gives the best results. The predictions obtained in the best test case of kNN-WAv with $k = 5$ are illustrated in Fig. 9. The figure shows that the predictions are

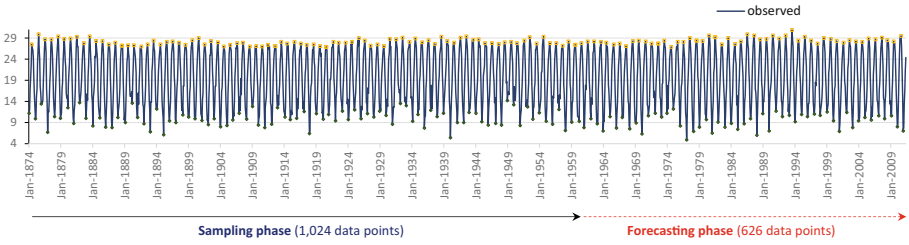

Fig. 8. Streaming time series *Temperatures at Savannah International Airport* and its major extrema

Table 4. The best test cases of each method

Method	MAPE	MAD	MSE
kNN-Av with $k = 4$	11.116	1.167	4.728
kNN-WAv with $k = 5$	**10.898**	**1.609**	**4.402**
SES with $\alpha = 0.5$	32.557	5	32.233
Hybrid with $\omega = 0.9$	10.982	1.837	5.799

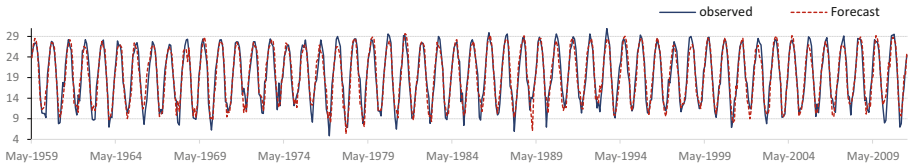

Fig. 9. The observations and predictions for *Temperatures at Savannah International Airport*

nearly the same observations although there are also many significant differences between the observation and the prediction at some important maxima.

6 Conclusions

The paper has introduced a new method to identify major extrema in streaming time series. The proposed method is a modification of the major extrema method [7], which works solely with static time series, to enable the task of finding major extrema in streaming time series. The proposed method is of low time complexity so it could detect major extrema quickly when a newly incoming data point of streaming time series has been just collected. Finding major extrema on the fly enables subsequent tasks of data mining on streaming time series to be performed efficiently and effectively such as determining correlation between streaming time series, segmenting streaming time series into subsequences more conveniently and sensibly for online forecasting, etc.

As for future work, we plan to use the proposed method to segment streaming time series for the task of subsequence join in streaming context.

Acknowledgement. This research is funded by Saigon University (SGU) under grant number CSA2022-06.

References

1. Rakthanmanon, T., et al.: Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2012), Beijing, China, pp. 262–270 (2012). https://doi.org/10.1145/2339530.2339576

2. Giao, B.C., Anh, D.T.: Similarity search for numerous patterns over multiple time series streams under dynamic time warping which supports data normalization. Vietnam J. Comput. Sci. **3**(3), 181–196 (2016). https://doi.org/10.1007/s40595-016-0062-4
3. Thuy, H.T.T., Anh, D.T., Chau, V.T.N.: Segmentation-based methods for top-k discords detection in static and streaming time series under euclidean distance. In: Cong Vinh, P., Rakib, A. (eds.) ICCASA 2021. LNICST, vol. 409, pp. 147–163. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-93179-7_12
4. Zhan, P., Sun, C., Hu, Y., Luo, W., Zheng, J., Li, X.: Feature-based online representation algorithm for streaming time series similarity search. Int. J. Pattern Recognit Artif Intell. **34**(5), 2050010 (2020). https://doi.org/10.1142/S021800142050010X
5. Giao, B.C., Anh, D.T.: An application of similarity search in streaming time series under DTW: online forecasting. In: Proceedings of the 8th International Symposium on Information and Communication Technology. Nha Trang, Vietnam, pp. 10–17 (2017). https://doi.org/10.1145/3155133.3155148
6. Fu, T.C., Chung, F.L., Luk, R., Ng, C.M.: Representing financial time series based on data point importance. Eng. Appl. Artif. Intell. **21**(2), 277–300 (2008). https://doi.org/10.1016/j.engappai.2007.04.009
7. Fink, E., Gandhi, H.S.: Compression of time series by extracting major extrema. J. Exp. Theor. Artif. Intell. **23**(2), 255–270 (2011). https://doi.org/10.1080/0952813X.2010.505800
8. Keogh, E., Smyth, P.: A probabilistic approach to fast pattern matching in time. In: Proceedings of 3rd International Conference Knowledge Discovery and Data Mining, California, USA, vol. 1997, pp. 24–30
9. Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proceedings of AAAI Workshop on Knowledge Discovery in Databases, Seattle, Washington, USA, pp. 359–370 (1994)
10. Giao, B.C: Time-series datasets. https://www.researchgate.net/publication/361923578_Time-series_datasets. Accessed 01 Jun 2022
11. Eurostat. Cows' milk collection and products obtained - monthly data. http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=apro_mk_colm. Accessed 31 Jul 2017
12. Hyndman, R.: Time series data library. https://datamarket.com/data. Accessed 01 Aug 2017