Detecting Variable Length Anomaly Patterns in Time Series Data

Ngo Duy Khanh Vy, Duong Tuan Anh
Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Vietnam
dtanh@cse.hcmut.edu.vn

Abstract— The time series anomaly detection algorithm, developed by Leng et al. (2008), can detect anomaly patterns of variable lengths in time series and hence does not require the length of the anomaly as an input parameter. This method consists of two stages: the first is segmenting time series; the next is detecting the anomaly patterns by calculating anomaly factor of each pattern and then judging whether a pattern is anomaly or not based on its anomaly factor. Since the lengths of patterns can be different from each other, the authors suggested that Dynamic Time Warping (DTW) distance should be used to calculate the distances between the patterns. Due to this suggestion, the algorithm leads to high computational complexity. In this paper, to improve the algorithm proposed by Leng et al., we apply homothetic transformation to convert every pair of patterns of different lengths into the same length so that we can easily calculate Euclidean distance between them. This modification accelerates the anomaly detection algorithm remarkably and makes it workable on large time series. Experimental results on eight time series datasets demonstrate the effectiveness of our proposed method in detecting anomaly patterns in time series.

Keywords— *time series; anomaly detection; segmentation; anomaly factor.*

I. INTRODUCTION

Recently, the problem of detecting unusual (abnormal, novel, deviant, anomalous, discord) subsequences in a time series has attracted much attention. Mining time series anomalies brings out the abnormal patterns embedded in a time series. Application areas that explore such time series anomalies include fault diagnostics, intrusion detection, fraud detection, auditing and data cleansing. Let us take an application example from the health care domain. The Electrocardiogram (ECG) can be considered as time series. It is an important procedure for doctors to analyze whether there is an anomaly pattern in a patient's ECG so as to determine the patient's health status.

Anomaly detection is a challenging topic, mainly because we need to obtain the lengths of anomaly patterns before detecting them. There has been an extensive study on time series anomaly detection in the literature. Some popular algorithms for time series anomaly detection include window-based methods such as brute-force and HOT SAX by Keogh et al. (2005) [5] and WAT by Bu et al. (2007) ([2]); a method based on neural-network by Oliveira et al. (2004) [11]; a method based on segmentation and Finite State Automata by Salvador and Chan (2005) [14]; a method based on time series segmentation and anomaly scores by Leng et al. (2008) [9]; a method based on PAA bit representation and clustering by Li et al. (2013) [10]; and a method which utilizes cluster-based outlier detection by Kha and Anh (2015) [7]. Most of these above-mentioned algorithms for time series anomaly detection ([6], [10], [14], [11]) require the user to specify the length of anomaly pattern as an input parameter, but this length is often unknown.

Among these algorithms, the method proposed by Leng et al. [9] is the first one in the literature that can detect anomaly patterns of variable lengths in time series and hence does not require the length of the anomaly pattern as a parameter supplied by user. This method consists of two stages. The first is segmenting time series using a quadratic regression model, finding out the longest and the shortest patterns with the lengths of l_{max} and l_{min} respectively. The next is discovering all the anomaly patterns by calculating anomaly factor of each pattern and then judging whether a pattern is anomaly or not based on its anomaly factor. Since in this algorithm, the lengths of patterns can be different from each other, Leng et al. suggested that Dynamic Time Warping (DTW) distance should be used to calculate the distances between the patterns in this algorithm. Due to this suggestion, the algorithm proposed by Leng et al. becomes a very complicated algorithm with high computational complexity and is not suitable to work on large time series.

In this work, to improve the algorithm proposed by Leng et al., we apply homothetic transformation to convert every pair of patterns of different lengths into the same length so that we can easily calculate Euclidean distance between them. In addition, we reduce the number of distance calculations in building the distance matrix by constraining the possible alignments between each pair of patterns much less than the value l_{max} - l_{min} + 1. These two modifications bring out a remarkable improvement for the original anomaly detection algorithm in time efficiency while maintaining the same detection accuracy. Besides, we try to apply in our proposed

anomaly detection algorithm another method of time series segmentation which is based on important extreme points rather than quadratic regression model.

Experimental results on eight real world time series datasets demonstrate the effectiveness of our proposed methods in detecting anomaly patterns of variable lengths in time series.

The rest of the paper is organizes as follows. In section 2 we give the background of the work. In section 3, we describe the proposed algorithms. In section 4 we show the experimental results. Finally, in section 5 we conclude our work and point out future work.

II. BACKGROUND

A. Some Definitions

Definition 1. *Time Series*: A time series $T = t_1, t_2, ..., t_m$ is an ordered set of m real values measured at equal intervals.

Definition 2. *Subsequence*: Given a time series T of length m, a subsequence C is a sampling of length $n < m$ of contiguous positions from T, i.e., $C = t_p, t_{p+1}, ..., t_{p+n-1}$, for $1 \le p \le m-n+1$. Sometimes, C is denoted as (s_p, e_{p+n-1}), where $s_p = t_p$ and $e_{p+n-1} = t_{p+n-1}$.

Definition 3. *Distance function*: $Dist(C, M)$ is a positive value used to measure the difference between two time series C and M, based on some measure method.

Definition 4. *k-distance of a pattern*: Given a positive integer k, a pattern set D and a pattern $P \in D$, the k-distance of P, denoted as k-$dist(P)$, is defined as the distance between P and a pattern $Q \in D$ such that.

i) For at least k patterns $Q' \in D$ it holds that $Dist(D, Q') \le Dist(D, Q)$.

ii) For at most k-1 patterns $Q' \in D \setminus \{Q\}$ it holds that $Dist(D, Q') < Dist(D, Q)$.

Definition 5. *Non-Self Match*: Given a time series T, its two subsequences P of length n starting at position p and Q starting at position q, we say that Q is a non-self-match to P, if $Dist(P, Q) \ge e$ or $|p - q| \ge n$, where e is a given value of distance threshold.

Definition 6. *Anomaly factor*: For any pattern set D and a pattern $P \in D$, k-$dist(D)$ denotes all k-$dist$ of patterns, anomaly factor of pattern P defined as the ratio of k-$dist(P)$ to $median(k$-$dist(D))$.

Definition 7. *Anomaly Pattern*: Given any pattern set D, a pattern $P \in D$, P is anomaly only if its anomaly factor is larger than a, where a is the threshold of anomaly pattern.

B. Detecting Variable Length Anomaly Patterns in Time Series based on Segmentation and Anomaly Factors

The following algorithm forms the background to the current work. Leng et al. [9] proposed the algorithm for detecting anomaly patterns in time series. This algorithm consists of two phases: segmenting time series based on quadratic regression model and detecting anomaly patterns based on anomaly factors.

Leng et al. used quadratic regression model to segment time series. Regression function is defined as $f(t) = \beta_0 + \beta_1 t + \beta_2 t^2$, where the t values, the values of time domain are real values. The segmentation algorithm searches for a segment $S = t_p, ..., t_{p+m-1}$ of a given time series T, starts at t_p and end at t_{p+m-1}, such that the sum of squared errors, is less than ε_1, where $f(t) = \beta_0 + \beta_1 t + \beta_2 t^2$ and ε_1 is a threshold specified by the user. We can extract all the patterns in the time series T by applying repeatedly this search procedure.

The segmentation procedure is described as follows.

1. Let $s_1 = 1$ denote the start position of the first segment, l denote the initial length of each segment, $m = s_1$ and then update this segment, the updating procedure is as follows. Calculate the values of β_0, $\beta_1 t$, and β_2 and the sum of squared errors:

$$SSE = \sum_{i=p}^{p+l-1} (f(i) - t_i)^2$$

2. If $SSE < \varepsilon_1$ let $l = l + 1$, go to step 1, else go to step 3.

3. Let (s_1, e_1) denote this segment, e_1 denotes the end position of this segment and $e_1 = l - 1$.

4. Let $i = 1$, if $Dist((s_1, e_1), (s_1 + i, e_1 + i)) \le \varepsilon_2$, increment i by 1, calculate it, repeat this procedure until $Dist((s_1, e_1), (s_1 + i, e_1 + i)) > \varepsilon_2$ or $i > (e_1 - s_1)$, let $s_2 = e_1 + i$, the s_2 is the start position of the next segment.

5. Let $m = s_2$, calculate e_2 using the steps 1-3, and then obtain the second segment (s_2, e_2).

6. Calculate s_j using the step 4, let $m = s_j$, calculate the value of e_j using the steps $1 - 3$, and then obtain the j-th segment (s_j, e_j).

7. Repeat the above procedure until the end position of the segment is n.

Notice that the segmentation procedure uses Dynamic Time Warping (DTW) as distance measure for the patterns and Step 4 aims to eliminate the trivial matches. The parameter ε_1 determines the length of each segment. The parameter ε_2 (called *non-self match threshold*) helps in removing trivial matches and hence it has impact on the number of extracted segments.

The procedure for Anomaly Pattern Detection is described as follows.

1. Find the maximum value, l_{max} and minimum value l_{min} of the lengths of all extracted segments.

2. Calculate the distance matrix $D = (d_{ij})_{m \times m}$ of segments:

$$d_{ij} = \min_{l_{min} \le l \le l_{max}} Dist((s_i, e_i), (s_j, s_j + l)) \quad (1)$$

where $1 \le i, j \le m$ and $i \ne j$.

3. Compute k-distance of each segments based on the distance matrix, let k-$dist(D)$ denote the set of all these distances.

4. Calculate median(k-$dist(D)$)

5. Calculate anomaly factor of each segment, and determine whether each segment is an anomaly pattern or not basing on the anomaly factor threshold a.

6. Given two anomaly patterns (s_i, e_i) and (s_j, e_j), if they are overlapped, merge them into one pattern.

Notice that in step 2, we need to compute ($l_{max} - l_{min} + 1$) DTW distances when calculating the DTW distance between (s_i, e_i) and (s_j, e_j), and the algorithm selects the minimum value of them as the real DTW distance of (s_i, e_i) and (s_j, e_j). Step 6 helps to merge two anomaly patterns if they are overlapped and this can weaken the influence of ε_l if the value of ε_l is smaller than its real value.

C. Dynamic Time Warping

Inspired by the need to handle time warping in similarity computation, Berndt and Clifford (1994) [1] introduced DTW distance to the data mining community, in order to allow a time series to be "stretched" or "compressed" to provide a better match with another time series. To compute the DTW distance between two time series, we have to use dynamic programming to solve an optimization problem and this solution method incurs a quadratic computational cost.

To speed up the DTW distance calculation, all practitioners using DTW apply some temporal constraint on the warping window size of DTW (Sakoe & Chiba, 1978 [13]) and/or utilize some lower-bounding techniques such as LB_Keogh (Keogh et al., 2002 [4]), and LB_Improved (Lemire, 2009 [8]). However, lower-bounding techniques often incur post-processing overhead while the exact DWT calculation is still unavoidable in practice. All these complications hinder the use of DTW distance in real world applications.

III. THE PROPOSED ALGORITHMS

The main idea of our method for detecting anomaly patterns of variable lengths in time series is to apply homothetic transformation to convert each pair of patterns of different lengths into the same length so that we can easily calculate Euclidean distances between them rather than using computationally complicated DTW distance. This idea was considered in our previous work which proposed an algorithm to discover time series motif based on significant extreme points and clustering (Truong et al., 2012 [15]). Now, this idea can be applied to the algorithm proposed by Leng et al. in order to accelerate its execution and make it suitable with large time series. We call our proposed method VL_QR|HT (Variable Length anomaly pattern detection based on Quadratic Regression and Homothety).

A. Homothetic Transformation and Modified Euclidean Distance

Homothety is a transformation in affine space. Given a point O and a value $k \neq 0$. A homothety with center O and ratio k transforms the point M to the point M' such that $\overrightarrow{OM'} = k \times \overrightarrow{OM}$.

Homothety can transform a time series T of length n ($T = \{y_1, y_2, ..., y_n\}$) to time series T' of length n' by performing the following steps. First, compute $Y\_MAX = MAX(y_1, ..., y_n)$, $Y\_MIN = MIN(y_1, ..., y_n)$. Second, set the center I of homothety with the coordinates $X\_C = n/2$, $Y\_C = (Y\_MAX + Y\_MIN)/2$. Next, perform the homothety with the center I and the ratio n'/n.

To compute Euclidean distance of two time series, first we check their lengths. If the two lengths are similar, we can compute their distance right away. Otherwise, we apply homothety to convert them to the same length and then compute their Euclidean distance. Here we select the average length of the two time series as the length to which we convert the two time series using homothety.

Fig.1. gives the outline of the algorithm that computes Euclidean distance of two time series of different lengths. The procedure *Homothety*(X, l) in this algorithm is for performing homothety to convert the time series X to the length l.

Function $Dist(X, Y)$
if ($X.length == Y.length$) **return** $Euclid(X, Y)$
else
       $MEAN = |(Y.length + X.length)| / 2$
       $Y' = Homothety(Y, MEAN)$
       $X' = Homothety(X, MEAN)$
return $Euclid(X', Y')$

Fig. 1. Outline of the algorithm that computes Euclidean distance using homothety.

Besides, we note that two 'similar' subsequences will not be recognized where a vertical difference exists between them. To make the Euclidean distance calculation in the anomaly pattern detection algorithm able to handle not only uniform scaling but also shifting transformation along vertical axis, we modify our Euclidean distance by using Minimum Euclidean Distance as a method of negating the differences caused through vertical axis offsets.

Given two subsequences: $T' = \{T'_1, T'_2, ..., T'_N\}$ and $Q' = \{Q'_1, Q'_2, ..., Q'_N\}$, the Euclidean distance between them is normally given by the following formula:

$$D(T', Q') = \sum_{i=1}^{N'} (T'_i - Q'_i)^2 \qquad (2)$$

In this work we use the Minimum Euclidean Distance which is defined by the following equation:

$$D(T', Q') = Min\left\{ \sum_{i=1}^{N'} (T'_i - Q'_i - b)^2 \right\} \quad \text{where } b \in \Re \quad (3)$$

From Eq. (3), we can derive a suitable value for the shifting parameter b such that we can find the best match between the two subsequences Q' and T' as follows:

$$b = \frac{1}{N'} \sum_{i=1}^{N'} (Q'_i - T'_i) \qquad (4)$$

Whenever given two subsequences *Q'* and *T'*, we have to compute the optimal value for the shifting parameter *b* using Eq. (4) before we can find the Minimum Euclidean Distance between them using Eq. (3).

B. *Reducing the Number of Distance Calculations in Building the Distance Matrix*

In the procedure for detecting anomaly patterns of variable lengths proposed by Leng et al. [9], to compute the distance between two patterns (s_i, e_i) and (s_j, e_j), we need to compute $(l_{max} - l_{min} + 1)$ times of distance calculations, where l_{min} and l_{max} are the maximum and minimum length of all the patterns extracted from the original time series (see Eq. 1 in Subsection 2.2). When l_{max} is much larger than l_{min}, the algorithm has to perform so many distance calculations in order to determine the real distance between two patterns and this might bring out the pathological cases in which a very short segment can match with a very long segment.

In order to reduce the number of distance calculations as well as to prevent pathological alignments between two patterns, in our proposed algorithm, we modify Eq. 1 by adding the parameter *r* and replace l_{max} and l_{min} with l_{upper} and l_{lower} respectively. The lower bound l_{lower} and the upper bound l_{upper} are defined by:

$$l_{upper} = \lfloor l_{avg}(1+r) \rfloor \qquad l_{lower} = \lceil l_{avg}(1-r) \rceil$$

where l_{avg} is the average length of all the segments extracted from the original time series. Now, the formula to compute the distance between two patterns *i* and *j* is:

$$d_{ij} = \min_{l_{lower} \leq l \leq l_{upper}} Dist((s_i, e_i), (s_j, s_j + l)) \qquad (5)$$

By the parameter *r*, we limit how far the length of the pattern *j* may differ from the average length of all the extracted patterns. The parameter *r* (called *length difference width*) should be chosen in order that the difference between l_{upper} and l_{lower} is not high while still maintaining the accuracy of the algorithm. The parameter *r* here has the same spirit of the Sakoe-Chiba band in constraining the warping path in DTW distance. Through experiment, we find out that *r* should vary in the range 0.05 to 0.25.

C. *Other Issue: Using Some Alternative Segmentation Method*

Using the same framework of VL_QR|HT, we can develop another anomaly detection algorithm by replacing the segmentation method based on quadratic regression with another time series segmentation method. The other segmentation method we selected here is based on the concepts of Important Extreme Points, proposed by Pratt and Fink [12]. This time series segmentation consists of two following steps.

1. Extract all important extreme points of the time series *T*. The result of this step is a sequence of extreme points $EP = (ep_1, ep_2, …, ep_l)$

2. Compute all the candidate patterns iteratively. A candidate pattern $CP_i(T)$, *i* = 1, 2,…, *l* -2 is the subsequence of *T* that is bounded by extreme points ep_i and ep_{i+2}. Candidate patterns are subsequences that may have different lengths. To be able to calculate distance between them, we bring them to the average length by using homothety.

After segmentation stage, we calculate anomaly factor for each pattern in the same way as in VL_QR|HT. We call this anomaly detection algorithm VL_EP|HT (Variable Length anomaly pattern detection based on Extreme Points and Homothety). Note that in identifying important extreme points of a time series, we need the parameter *R*, called *compression rate*, which is greater than one and an increase of *R* leads to selection of fewer important extreme points. To set a lower bound for the length of all extracted subsequences, we introduce the parameter *min_length*, the minimum time lag between two adjacent important extreme points. When a candidate pattern is extracted, its length must be greater than or equal to 2*min_length. This constraint is necessary to ensure that the segmentation phase does not bring out too short subsequences.

IV. EXPERIMENTAL EVALUATION

We implemented the comparative methods with Microsoft Visual C# and conducted the experiments on an Intel® Core™ 2 Duo 2.0GHz, Ram 3072MB PC. We implemented all four algorithms: VL_QR|DTW(the original algorithm proposed by Leng et al.), VL_QR|HT, VL_EP|HT and HOT SAX. The first experiment aims to compare VL_QR|HT to original VL_QR|DTW in terms of time efficiency. The second experiment aims to compare VL_QR|HT and VL_EP|HT to HOT SAX in terms of time efficiency and anomaly detection accuracy.

TABLE I. Parameter values in the three algorithms for each dataset

Datasets	VL_QR| HT	VL_EP| HT	HOT SAX
ECG108 (17500 points)	ε_1= 5.0, ε_2 = 0.3, a = 3.5, r = 0.1	R = 1.04, *min_lenth* = 50, a = 4, r = 0.1	n = 600, *PAA_size* = 60
ECG308 (1300 points)	ε_1= 51000, ε_2 = 50, a = 2.2, r = 0.1	R = 1.03, *min_lenth* = 10, a = 2.5, r = 0.23	n = 60, *PAA_size* = 6
ERP (5000 points)	ε_1 = 3.0, ε_2 = 1.0, a = 3.0, r = 0.15	R = 1.42, *min_lenth* = 10, a = 3.5, r = 0.1	n = 100, *PAA_size* = 10
Memory (6875 points)	ε_1= 8.0, ε_2 = 0.1, a = 2.2, r = 0.1	R = 1.1, *min_lenth* = 40, a = 1.6, r = 0.1	n = 100, *PAA_size* = 20
Power Demand Italy (7000 points)	ε_1 = 100000, ε_2 = 100, a = 2.5, r = 0.1	R = 1.8, *min_lenth* = 20, a = 3.0, r = 0.1	n = 300, *PAA_size* = 30
Dutch Power Demand (9000 points)	ε_1 = 31000000, ε_2 = 500, a = 1.6, r = 0.2	R = 1.7, *min_lenth* = 180, a = 1.33, r = 0.25	n =1200, *PAA_size* = 120
Stock20 (5000 points)	ε_1= 2.0, ε_2 = 0.1, a = 3.5, r = 0.1	R = 1.01, *min_lenth* = 200, a = 1.5, r = 0.1	n = 700, *PAA_size* = 70
TEK1 (5000 points)	ε_1 = 14.5, ε_2 = 0.1, a = 2.0, r = 0.1	R = 1.02, *min_lenth* = 50, a = 1.5, r = 0.1	n = 128, *PAA_size* = 8

Our experiments were conducted over the datasets from the UCR Time Series Data Mining Archive for discord discovery [6]. There are 8 datasets used in these experiments. The datasets are from different areas (finance, medicine, manufacturing, science). The names and lengths of the eight datasets are given in Table I.

For each dataset, we have to set the required parameters for each of the three algorithms: VL_QR|HT, VL_EP|HT and HOT SAX. For VL_QR|HT, the parameters are regression error threshold ε_1, non-self match threshold ε_2, anomaly factor threshold a and the length difference width r. For VL_EP|HT, the parameters are compression rate R, the minimum length of extracted subsequence min_length, anomaly factor a and the length difference width r. For HOT SAX, the parameters are the discord length n and the length of PAA segment PAA_size. The values of parameters in the three algorithms for each dataset are shown in Table I.

A. Experiment 1: Comparing VL_QR |HT to VL_QR |DTW

This experiment aims to compare the efficiency of our proposed algorithm, VL_QR| HT to that of the algorithm proposed by Leng et al. (VL_QR|DTW).

Over 8 datasets, we found out that the anomaly pattern detected by VL_QR|HT is exactly the same as the one detected by VL_QR|DTW.

Table II shows the run times (in seconds) of the two algorithms over 8 datasets. Since VL_QR|DTW performs extremely slowly, we have to limit the lengths of the eight datasets to less than 3000 data points. Besides, to accelerate the DTW distance calculation, we employed a multithreading method, proposed by Huy in 2015 [3], in computing this distance.

The experimental results in Table II demonstrate that with all the datasets, the VL_QR|HT is remarkably faster than VL_QR|DTW while brings out the same accuracy. The speedup of our proposed algorithm over to the original algorithm by Leng et al. varies from 4 (dataset Memory) to 97 times (dataset Stock20) and in average is about 33.6.

TABLE II. Run times of VL_QR|HT and VL_QR|DTW over eight datasets

Dataset	VL_QR|HT	VL_QR|DTW
ECG 108 (1000 points)	1	14
ECG 308 (1300 points)	1	11
ERP (1000 points)	1	5
Memory (1000 points)	1	4
Power Demand In Italy (3000 points)	6	200
Dutch Power Demand (3000 points)	1	24
Stock20 (3000 points)	2	194
TEK16 (2000 points)	1	81

B. Experiment 2: Comparing VL_QR|HT and VL_EP|HT to HOT SAX

In the second experiment we compare VL_QR|HT and VL_EP|HT to HOT SAX in terms of time efficiency and anomaly detection accuracy. The HOT SAX [5] is used as the baseline algorithm in this experiment. The HOT SAX is selected for comparison due to its popularity. It is the most cited algorithm for detecting time series discords up to date and has been applied in many applications.

In order to compare with HOT SAX which always brings out the top-anomaly pattern in a time series, in this experiment, we set the parameter k in *k-distance* for patterns to 1 for both VL_QR|HT and VL_EP|HT.

Table III shows the experimental results about the anomaly patterns found on each of 8 time series datasets by the three algorithms. Over 8 datasets, we found out that the anomaly pattern detected by VL_QR|HT or VL_EP|HT is exactly the same as the one detected by HOT SAX.

However, for each dataset there is some difference between the start position of the anomaly pattern found by VL_QR|HT or VL_EP|HT to that of the same anomaly pattern found by HOT SAX. To compute location difference, we use the quantity d defined as $d = |p - q|/l \times 100$, where p is the start position of the anomaly pattern found by our proposed algorithm and q is the start position of the anomaly pattern found by HOT SAX and l is the length of the found anomaly pattern.

From Table III, we found out that the location difference between VL_QR|HT and HOT SAX is smaller than the location difference between VL_EP|HT and HOT SAX in 5 out of 8 datasets. This phenomenon indicates that VL_QR|HT can detect time series anomaly patterns with slightly higher accuracy than VL_EP|HT.

TABLE III Positions of anomaly patterns detected by VL_QL|HT and VL_EP|HT and HOT SAX

Dataset	VL_QR| HT		VL_EP| HT		HOT SAX
	Start position	Location difference (*d*)	Start position	Location difference (*d*)	Start position
ECG108	10875	1.5%	10792	12.3%	10866
ECG308	656	33%	673	5%	676
ERP	2617	32%	2602	17%	2585
Memory	2425	2%	2399	11%	2421
Power Demand Italy	5311	4%	5383	13.33%	5343
Dutch Power Demand	6424	1.3%	6277	11%	6466
Stock20	2961	1.7%	2960	1.6%	2949
TEK16	4262	24%	4197	27%	4231

Additionally, we measured the execution times (in seconds) of the three algorithms over 8 datasets. From the experimental results in Table IV we can see that

- VL_QR|HT and VL_EP|HT perform remarkably faster that HOT SAX in all the tested datasets.
- The speedup of VL_QR|HT over HOT SAX varies from 8 to 94 times and the speedup of VL_EP|HT over HOT SAX varies from 6 to 94 times.
- The run time of VL_QR|HT is always lower than that of VL_EP|HT over all the datasets.

TABLE IV. Run times of VL_QR|HT, VL_EP|HT and HOT SAX over eight datasets

Dataset	VL_QR| HT	VL_EP| HT	HOT SAX
ECG108	10	47	458
ECG308	0.31	2	13
ERP	7	14	60
Memory	4	6	79
Power Demand Italy	9	11	102
Dutch Power Demand	1	9	302
Stock20	1	1	94
TEK16	1	1	52

V. CONCLUSIONS

We have introduced two proposed algorithms which are improved variants of the method for variable length anomaly pattern detection in time series proposed by Leng et al. [9]. These two algorithms, called VL_QR|HT and VL_EP|HT, hinge on using homothetic transformation to convert each pair of patterns of different lengths into the same length so that we can easily calculate Euclidean distances between them. This modification that avoids using DTW distance brings out a remarkable improvement for the original algorithm in time efficiency without compromising anomaly detection accuracy. Experimental results on eight datasets demonstrate that our proposed algorithms outperform the VL_QR|DTW in time efficiency. Efficiency is achieved because Euclidean distance is much faster to compute than DTW distance. Meanwhile some issues brought out by VL_QR|HT and VL_EP|HT are still open. It is not easy to estimate the regression error threshold ε_1 and the non-self match threshold ε_2 in the segmentation phase which have impact on the speed of the anomaly detection phase of VL_QR|HT. It is also not easy to find the suitable value for the compression rate R in identifying important extreme points which influences on the accuracy of the anomaly detection of VL_EP|HT.

As for future work, we plan to apply some more advanced method of time series segmentation to our method in order to detect anomaly patterns of variable lengths in time series more efficiently.

REFERENCES

[1] D. Berndt, J. Clifford, "Finding patterns in time series: a dynamic programming approach", Journal of Advances in Knowledge Discovery and Data Mining, AAA/MIT Press, Menlo Park, CA, pp. 229-248, 1996.

[2] Y. Bu, T.W. Leung, A. Fu, E. Keogh, J. Pei, and S. Meshkin, "WAT: Finding top-K discords in time series database", Proceedings of the 2007 SIAM International Conference on Data Mining (SDM'07), Minneapolis, MN, USA, April 26-28, 2007.

[3] V. T. Huy, "Anytime k-medoids clustering of time series under dynamic time warping using an approximation technique", Master Thesis, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam, 2015.

[4] E. Keogh, "Exact indexing of dynamic time warping", Proceedings of 28th International Conference on Very Large Data Bases , Hong Kong, pp. 406-417, 2002.

[5] E. Keogh, J. Lin, and A. Fu, "HOT SAX: efficiently finding the most unusual time series subsequence", Proc. of 5th ICDM, Houston, Texas, pp. 226–233, 2005.

[6] E. Keogh, www.cs.ucr.edu/~eamonn/discords/ (accessed on January 24 - 2015).

[7] N.H. Kha and D. T. Anh, "From cluster-based outlier detection to time series discord discovery", In: Trends and Applications in Knowledge Discovery and Data Mining – PAKDD 2015 Workshops: Big_PMA, VLSP, QIMIE, BAEBH, Ho Chi Minh City, Vietnam, May 19-21, X. L. Li et al. (Eds.), LNAI 9441, Springer, pp. 16-28, 2015.

[8] D. Lemire, "Faster retrieval with a two-pass dynamic-time-warping lower bound", Pattern Recognition, vol. 42, no. 9, pp. 2169-2180, 2009.

[9] M. Leng, X. Chen and L. Li, "Variable length methods for detecting anomaly patterns in time series", Proc. of Int. Symposium on. Computational Intelligence and Design (ISCID'08), Vol. 2, 2008.

[10] G. Li, O. Braysy, L. Jiang, Z. Wu, Y. Wang, "Finding time series discord based on bit representation clustering", Knowledge-Based Systems, vol.52, pp. 243-254, 2013.

[11] A. L. I. Oliveira, F.B.L. Neto, and S.R. L. Meira., "A method based on RBF-DAA neural network for improving Novelty detection in time series", Proc. of 17th International FLAIRS Conference, AAAI Press, Miami Beach, Florida, USA, 2004.

[12] K. B. Pratt and E. Fink, "Search for patterns in compressed time series", International Journal of Image and Graphics , vol. 2, no. 1, pp. 89-106, 2002.

[13] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", IEEE Trans. Acoustics, Speech, and Signal Proc., vol. ASSP-26, pp. 43-49, 1978.

[14] S. Salvador, P. Chan, "Learning states and rules for time series anomaly detection", Applied Intelligence, vol. 23, no.3, pp. 241 -255. 2005.

[15] C. D. Truong, H. N. Tin, D. T. Anh, " Combining motif information and neural network for time series prediction", Int. Journal of Business Intelligence and Data Mining, vol. 7, no. 4, pp. 318-339, 2012.