Journal of Computers Vol. 30 No. 2, 2019, pp. 68-83
doi:10.3966/199115992019043002006

Chiang Mai Earthquakes Prediction: A Dynamic Time Warping Embed the Polynomial Contraint

Suppavich Kongthanarit[1*], Prasong Praneetpolgrang[2], Jigyasa Bajracharya[3], Somrak Petchartee[4]

[1] Department of Computer Engineering, Chulalongkorn University, Bangkok, Thailand
flukyysupp@gmail.com

[2] Department of Information Technology, Sripatum University, Bangkok, Thailand
prasong.pr@gmail.com

[3] Department of Electronics and Communications Engineering, Thammasat University, Kathmandu, Nepal
jigyasa.bajra@gmail.com

[4] Brownien Laboratory, Innovation Center, CAT Telecom, Thailand
somrak.p@cattlecom.com

Received 15 December 2018; Revised 15 February 2019; Accepted 15 February 2019

Abstract. Earthquakes are sudden natural disasters that endanger people's lives and properties all around the world. Although much work has been done in the field of prevention of such damages via timely prediction the work needs to be tailored to the area-specific dataset for optimum accuracy. With the time series seismic event data of Chiang Mai for 4 years, we focus our study to provide a prediction of future earthquakes here by using z-normalization and a polynomial constraint in DTW on CUDA-enabled accelerators. By comparing polynomial constraint to Sakoe-Chiba Band as well as Itakura band, we have verified our work to have reduced the percentage of computational time and improved accuracy. The aim of this paper is to demonstrate a prediction accuracy of 0.95.

Keywords: Itakura Band, polynomial constraint, Sakoe-Chiba Band, time series, Z-normalization

1 Introduction

Natural disasters are a destructive phenomenon that occurs beyond our control. They are catastrophic and not only take thousands of lives but also cause a lot of financial damage in properties. Earthquakes are one of several such geological disasters. Since these calamities are so spontaneous and damaging, the only way to defend ourselves is to be prepared. It is of the utmost importance for the safety of human life to be able to mitigate the damages by being well informed. Therefore, in this paper we focus on the prediction of earthquakes.

These days with inventions such as geophones, seismometers and accelerometers, it is easy to record seismic data. However, it is important to figure out how to efficiently utilize such data for profitable results. When it comes to prediction, one of the most reliable methods is to use time series data. It is an orderly collection of data sets pertaining to many possible measurements. Most of the current time series recognition systems are based on the pattern comparison of whole words. For example, in acoustic analysis, speech is represented as a temporal sequence of feature vectors that is compared with the stored comparison patterns of all the words in the vocabulary using an appropriate distance metric. One of the major problems with the classification of words is the distortion of timing. Time series recognition is the basis of time series mining and becomes one of the most studied areas in the time series mining literature [1].

---

* Corresponding Author

A word does not have to be pronounced by different people to have the same meaning. The problem of time coordination is seen to be solved by dynamic programming, which can be used to determine the best assignment of the time axes of an unknown word to each comparison pattern.

In our case, we used the datasets available from the Incorporated Research Institutions for Seismology (IRIS) database collected by using data from an accelerometer. The only unit that we considered is acceleration. Our aim is to determine the patterns seen in previous earthquakes in order to train a system to be able to identify the symptoms well in advance. In order to do so, finding similarities between two time series is a necessary step. Similar approaches can be seen in other data mining applications, in fields such as medical, business or scientific work.

We propose the use of Dynamic Time Warping algorithm for our time series data along with the introduction of a polynomial constraint, not only for forecasting abrupt earthquakes but also the many aftershocks that are sure to follow. We intend to implement a fully functional earthquake prediction system, which could prove to be useful in many future fields of research as well.

2 Related Works

In the course of our research for previous work relating to this field, we found several different approaches pertaining to the prediction of earthquakes. For instance, [2] employs the use of soft computing. In their work, they attempt to make a comparison in the use of non-linear time series data via multiple methods. Similar to our work, they considered only the earthquake acceleration value from their dataset. They utilized Takens's theorem in order to embed the time series data and calculate the time lag accordingly. After this, they deduced the dimension of the state space by selecting the dimension that showed the most chaotic behavior. They concluded that Neural Network and GMDH methods were the fastest in prediction, however, they could not present good results for any data with highly chaotic characteristics such as logistic mapping.

In [3], we can see a different approach via FFT (Fast Fourier Transform). Their research involved monitoring vibrations from various ranges, such as from the hypocenter to the selected area for prediction with normalized amplitude. Next, the waveforms were done by using cross-correlation and with the use of FFT, vibration is predicted. However, the drawback of this method was that the FFT computation uses a lot of the processor time. Following further work in early earthquake prediction is [4]. Here, they started with the extraction of six representative predictive indexes in terms of time, space and intensity. Then, they used a three-layer BP neural network for the prediction modeling with parameter optimization based on PSO and IPSO algorithm. Finally, they compared the training based on the training convergence.

While there are many methods to deal with time series data, we chose dynamic time warping as we found that the similarity matching feature was the most suitable for our work. In [5], we can see an improved DTW algorithm by adapting the Sakoe-Chiba band into the parallelogram band. It is done with the aim to reduce computation time which is also one of the features of our work. In the same vein is [6], with the use of DTW in Pattern Recognition of ECG Changes in Heart Rhythm Disturbances. They used DTW for electrocardiograms (ECGs) changes, which is a powerful algorithm in terms of precision but takes too much time to run through the data without constraint. Therefore, we chose to introduce a polynomial constraint to improve the accuracy while handling our time-series data by using DTW to reduce computation time.

3 Dataset

The dataset used in this paper has been taken from the Incorporated Research Institutions for Seismology (IRIS). The data is in SEED format version 2.4. It has been collected from the station IU CHTO (Chiang Mai, Thailand) with 3 channels (HH1, HH2, and HHZ). The seed volume spans four years i.e. from 2014, August 16, 07:02:34 to 2018, August 6, 22:16:15. In total, it contains 221,030,314 data points. The data is collected with the epi-sensor accelerometer model FBA_ES-T. In SEED format, it has the structure as shown below:

In Fig. 1, the SEED file consists of 5 layers. First, the volume index control header contains information about the time of the data, logical record length, and format version of the logical volume.

Next, the abbreviation dictionary control header contains the definitions of abbreviations used in the other control headers. Following that, the station control headers contain the operating characteristics for a station as well as its channels. Next, the Time span control headers contain the time span in which the time series data are recorded. Finally, it consists of the time series data within those time spans.

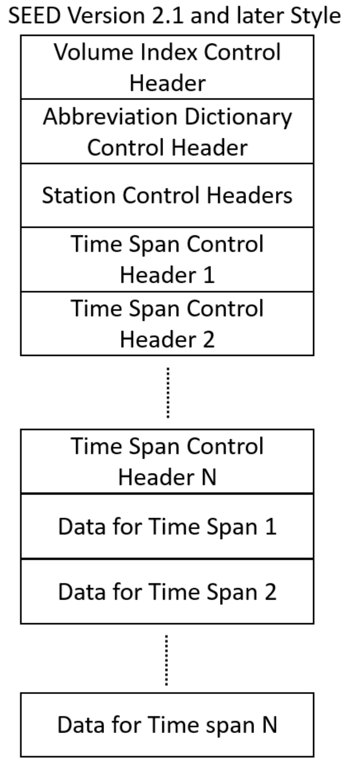
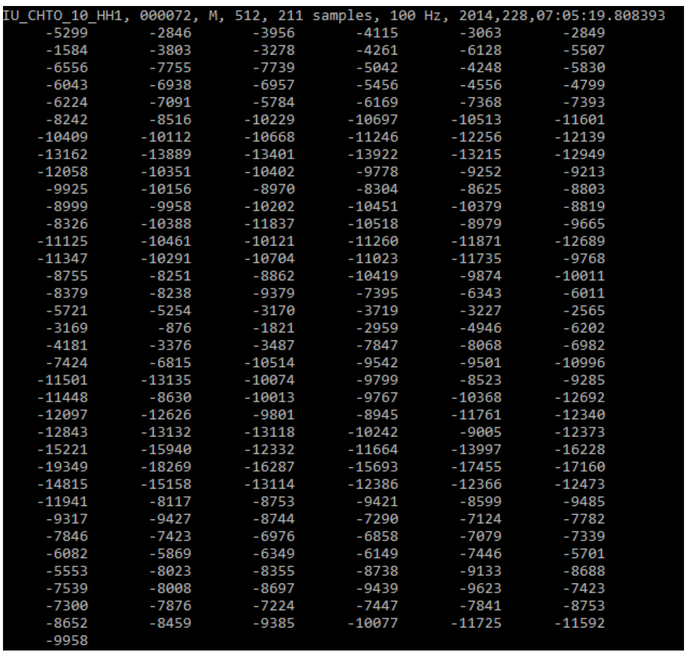


Fig. 1. Data structure of SEED file [9]

In Fig. 2, samples are read with libmseed library. The first row shows the headers consisting of the station control header and the time span control header. It provides auxiliary information about the volume, the station-channels, and the data of each time spans. The raw time series data that we used follows the headers.



```
IU_CHTO_10_HH1, 000072, M, 512, 211 samples, 100 Hz, 2014,228,07:05:19.808393
     -5299       -2846       -3956       -4115       -3063       -2849
     -1584       -3803       -3278       -4261       -6128       -5507
     -6556       -7755       -7739       -5042       -4248       -5830
     -6043       -6938       -6957       -5456       -4556       -4799
     -6224       -7091       -5784       -6169       -7368       -7393
     -8242       -8516      -10229      -10697      -10513      -11601
    -10409      -10112      -10668      -11246      -12256      -12139
    -13162      -13889      -13401      -13922      -13215      -12949
    -12058      -10351      -10402       -9778       -9252       -9213
     -9925      -10156       -8970       -8304       -8625       -8803
     -8999       -9958      -10202      -10451      -10379       -8819
     -8326      -10388      -11837      -10518       -8979       -9665
    -11125      -10461      -10121      -11260      -11871      -12689
    -11347      -10291      -10704      -11023      -11735       -9768
     -8755       -8251       -8862      -10419       -9874      -10011
     -8379       -8238       -9379       -7395       -6343       -6011
     -5721       -5254       -3170       -3719       -3227       -2565
     -3169        -876       -1821       -2959       -4946       -6202
     -4181       -3376       -3487       -7847       -8068       -6982
     -7424       -6815      -10514       -9542       -9501      -10996
    -11501      -13135      -10074       -9799       -8523       -9285
    -11448       -8630      -10013       -9767      -10368      -12692
    -12097      -12626       -9801       -8945      -11761      -12340
    -12843      -13132      -13118      -10242       -9005      -12373
    -15221      -15940      -12332      -11664      -13997      -16228
    -19349      -18269      -16287      -15693      -17455      -17160
    -14815      -15158      -13114      -12386      -12366      -12473
    -11941       -8117       -8753       -9421       -8599       -9485
     -9317       -9427       -8744       -7290       -7124       -7782
     -7846       -7423       -6976       -6858       -7079       -7339
     -6082       -5869       -6349       -6149       -7446       -5701
     -5553       -8023       -8355       -8738       -9133       -8688
     -7539       -8008       -8697       -9439       -9623       -7423
     -7300       -7876       -7224       -7447       -7841       -8753
     -8652       -8459       -9385      -10077      -11725      -11592
     -9958
```

Fig. 2. Samples of accelerometer data during a span of time

4 Algorithms

Following are the detailed algorithms utilized in this project:

4.1 Z-normalization

Z-normalization [10] is a necessary pre-process before DTW. In most cases, the results from DTW will be very poor without z-normalization. As you can see in Fig. 3, if you look at the two time series before z-normalization, it is obvious that they have very similar patterns and also almost the same mean and S.D. However, when DTW is applied to these series, the resulting match is completely incorrect as without z-normalization, DTW will automatically choose the nearest distance despite the inaccurate distances in real practice.

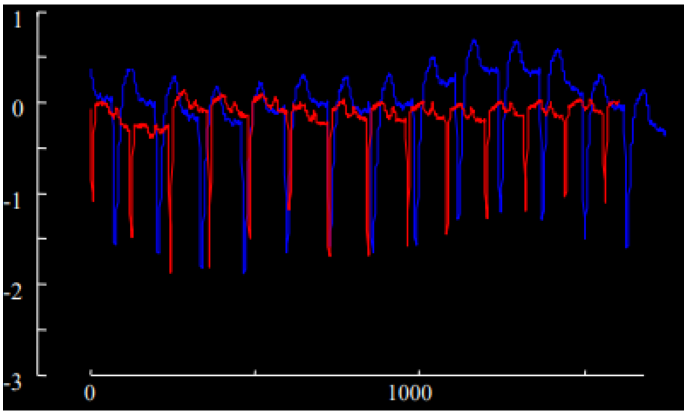


Fig. 3. Example of problem without z-normalization [11]

$$x_i' = \frac{x_i - \mu}{\sigma}, i \in N \quad \textbf{(1)}$$

From equation (1), $x_i = \{x_1, x_2, x_3, \ldots x_n\}$ where $i \in N$ is a set of time series containing n data.

μ is the mean of each elements in series
σ is the Standard Deviation (S.D.) of the same.
Following this, each element from the series will transform into a new series with the mean getting closer to 0 and S.D. getting closer to 1.

As observed in Fig. 4 and Fig. 5 before normalization, the two sequences are very different. The reason we apply normalization to both series is to focus on the structural similarities and dissimilarities. Hence, after normalization the second series is amplified to become highly similar to the first one and can be studied easily.

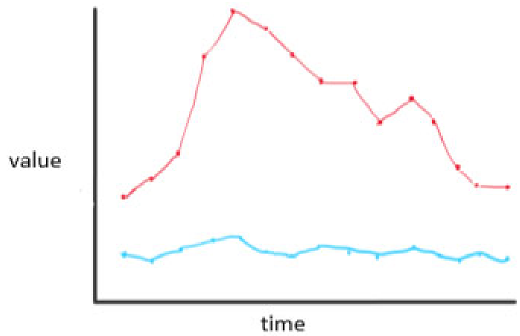


Fig. 4. Graph comparing two time series sequences before normalization [10]

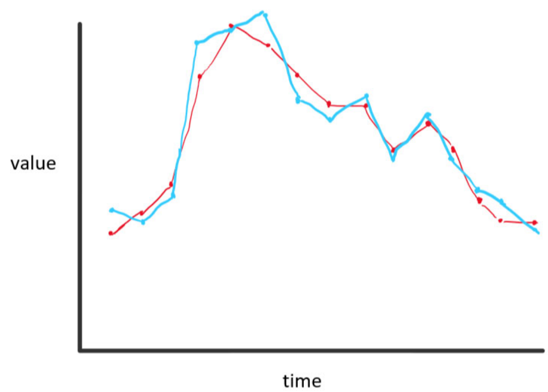


Fig. 5. Graph comparing two time series sequences after normalization [10]

4.2 Dynamic Time Warping

To analyze time series, Dynamic Time Warping (DTW) [8] is one of the popular approaches as it deals with the measurement of similarities for different time series sequences which may vary in terms of time. For instance, two different speeds of men walking are given in Fig. 8 in the form of time series data. They walk in almost the same pattern but you can't detect the similarity with the Euclidean distance method because, they do not maintain the same distance between the two series. The different temporal structuring of the two time series productions is reflected in the relative distortion of their time scales, which is shown in Fig. 6 and Fig. 7. Unfortunately, the measuring distance with Euclidean may be a fragile distance measure if the point stream exhibits local shifts and scaling along the measurement axis [12].

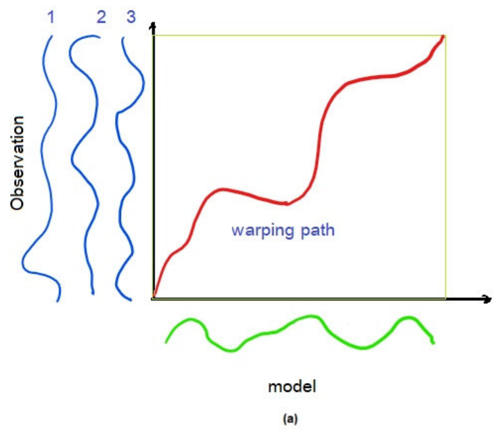


Fig. 6. Non-linear time distortion between time series

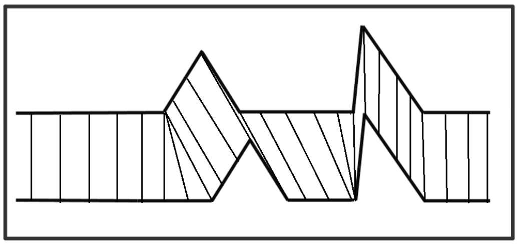


Fig. 7. Dynamic time warping

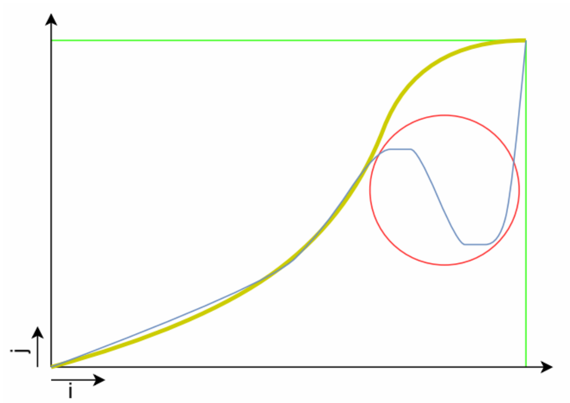


Fig. 8. Monotonicity in warping function

To solve the problem, DTW is introduced into time series matching which is widely used in speech signal DTW algorithm is based on Dynamic Programming techniques as describes in [14]. Fig. 7 shows the example of how one times series is "warped" to another [15].

The path represents a presumed assignment between times of corresponding time series and is called warping function. The aim is to make predictions as precisely and quickly as possible. The more distorted the warping function is, the more difficult the prediction and therefore the higher the error rate. Furthermore, there are an incredible number of possible paths that slow down the prediction. For this reason, the paths must first be examined for certain requirements.

Algorithm 1. Pseudo code of dynamic time warping algorithm [7]

```
DTW (v1, v2) {
// where the vectors v1=(a1, ..., an),
v2=(b1, ..., bm) are the time series
with n and m
time points
      Let a two dimensional data
matrix S be the store of similarity
measures such that
S[0, ..., n 0, ..., m], and i, j, are
loop index, cost is an integer.
      // initialize the data matrix
      S[0, 0] := 0
      FOR i := 1 to m DO LOOP
            S[0, i] := ∞
      END
      FOR i := 1 to n DO LOOP
            S[i, 0] := ∞
      END
      // Using pairwise method,
incrementally fill in the similarity
matrix with the
      differences of the two time
series
      FOR i := 1 to n DO LOOP
            FOR j := 1 to m DO LOOP
            // function to measure the
distance between the two points
                  cost := d (v1[i],
v2[j]) where d is the difference
between points
                  S[i, j] := cost +
MIN (S[i-1, j],   // increment
S[i, j-1],        // decrement
S[i-1, j-1])      // match
            END
      END
      Return S[n, m]
}
```

We utilized this property on our dataset in order to find the minimum path by measuring the similarities for our time series sequences. It is capable of finding an optimal match despite the sections being stretched or compressed.

The steps of DTW algorithm (1) includes first creating a mapping between two vectors (time series) with size (M, N), where M and N are the lengths of vectors. Then, we fill the distance between each possible elements of both vectors. Every warping path between the two time series will run completely through the matrix but we only want the best (shortest) one. Hence, we use the backtracking procedure to find the minimum distance and select that path as the optimal warping path.

We can speed up the computation by ignoring some of outer warping paths. The basic restrictions on the warping function are as follows:

Monotonicity. The monotonicity ensures that the characteristics which have already been considered are not taken up again during the process, as shown below in Fig. 8. The behavior shows up as a descent for the j-values in the alignment. It must follow the following conditions:

$$i_{s-1} \leq i_s \quad \textbf{(2)}$$

$$j_{s-1} \leq j_s \quad \textbf{(3)}$$

Each segment of the warping path has to increment at least one index of either data sets. As a result, DTW is not allowed to map an index tuple several times.

Continuity. Continuity of the warping function ensures that there are no gaps in the alignment. The values of the model are thus taken and compared at equal intervals. It safeguards against the loss of important information. This behavior is shown in Fig. 9. It must follow the following conditions:

$$i_s \leq i_{s-1} \leq 1 \quad \textbf{(4)}$$

$$j_s \leq j_{s-1} \leq 1 \quad \textbf{(5)}$$

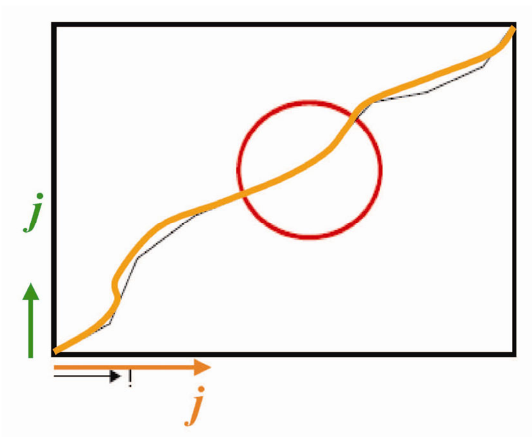


Fig. 9. Continuity in warping function

Consecutive nodes in point must be reached by horizontal, vertical or diagonal steps of length 1. Hence, DTW matches every index of both data sets without any gaps.

Starting point. To ensure that the complete time series is considered for analysis, it must start at the point (1, 1) and end at the point (n, m). This can be seen in Fig. 10 and can be defined by the following equations:

$$i_1 = 1, \quad i_k = n \quad \textbf{(6)}$$

$$j_1 = 1, \quad j_k = m \quad \textbf{(7)}$$

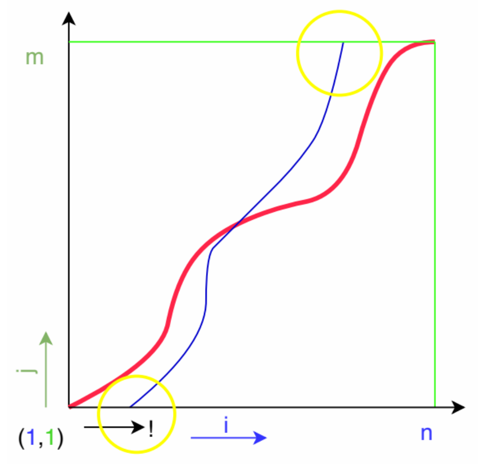


Fig. 10. Boundary conditions in warping function

It is also possible for important characteristics to be skipped during the classification. Such behavior can be seen in the alignment as soon as it leaves the warping window (Fig. 11). This process can be defined by the following:

$$|i_s - j_s| \leq r \quad \textbf{(8)}$$

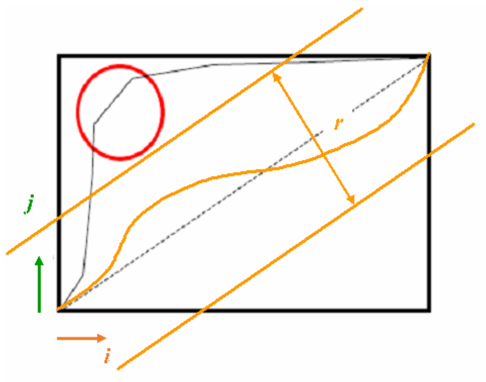


Fig. 11. Window in warping function

From equation (8), where $\boldsymbol{r} > 0$ is the window length.

Alignments. Finally, features from the similarity comparisons can also be reflected by too steep or too shallow alignments (Fig. 12). This shows that the short sequence sections are matched to long sections (Fig. 12). This behaviour must be prevented by means of a clever algorithm. The algorithm can be expressed as follows:

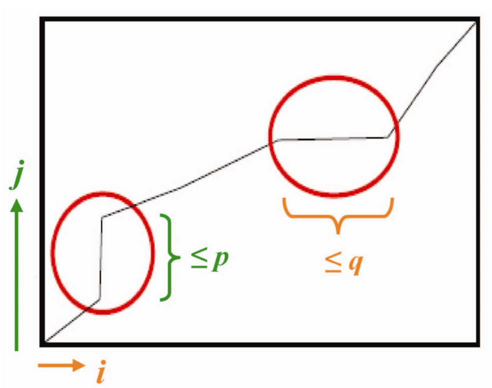


Fig. 12. Slope constraint in in warping function

$$\frac{j_{sp} - j_{s0}}{i_{sp} - i_{s0}} \leq p \tag{9}$$

$$\frac{i_{sp} - i_{s0}}{j_{sp} - j_{s0}} \leq q \tag{10}$$

From equation (9) and equation (10), where $\boldsymbol{q} \geq 0$ is the number of steps in the $\boldsymbol{x}$-direction and $\boldsymbol{p} \geq 0$ is the number of steps in the $\boldsymbol{y}$-direction.

After $\boldsymbol{q}$ steps in $\boldsymbol{x}$, one must step in $\boldsymbol{y}$ and vice versa:

$$S = \frac{p}{q}, \quad S \in [0, \propto] \tag{11}$$

4.1 Polynomial Constraint DTW

A common DTW optimization is to add constraint conditions on the warping paths. Those constraints do not only speed up DTW computations but also prevent alignments by controlling the route of a warping path.

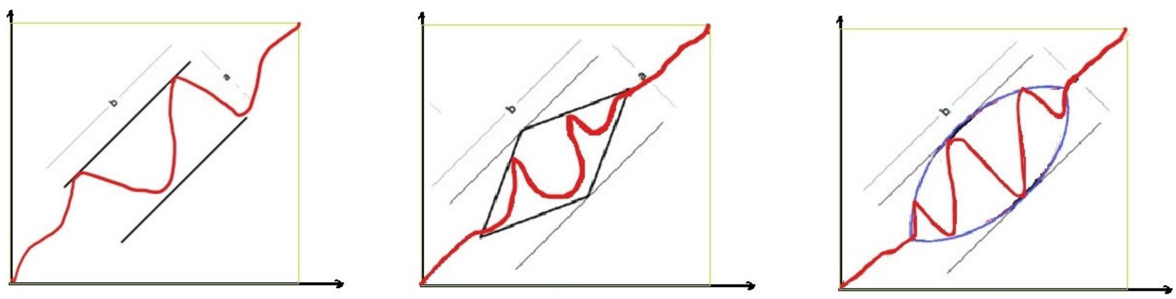
To speed up the similarity search, fixed limits can be created. Although the DTW algorithms achieve a drastic reduction in computing effort compared to the complete calculation of all possible paths, the computing power to be provided can be considerable, especially when comparing an unknown time series with many possible candidates. Any savings in calculation that do not impair the accuracy of the recognition result is therefore desirable.

The sequences are to be deleted as soon as they leave the boundaries. In this paper, different functions for the limitation are considered and compared in their quality. All functions have high demands on the CPU and I/O time.

A simple limiting function is the Sakoe-Chiba Band. As can be seen in Fig. 13(a), this boundary window consists of two parallel diagonals with a boundary distance *a*. If the path is within the window, it is considered for the similarity analysis. One problem with this boundary window is that it does not take the start and end points into account the requirements for the boundary conditions.

The described problem can be solved with the Itakura parallelogram. The idea is shown in Fig. 13(b). The peak of the parallelogram is fixed in the middle. Thus, all paths have the same length.

Another idea that we proposed is one which takes the boundary conditions into account, i.e. the polynomial boundary window which is illustrated in Fig. 13(c). The function is symmetrical to the diagonal and resembles a parabola. The window function starts and ends at the boundary points. Similarly, a performance evaluation for the polynomial constraint used is included in Fig. 14.



(a) The Sakoe-Chiba constraint

(b) Itakura constraint

(c) Polynomial constraint based different curvatures respectively with a-miner and b-major length

Fig. 13. The boundary window in different functions

When a warping path exceeds the Itakura band, the method will abort the similarity search and discard the time series. The result would be negative even though this path is very similar to the model. However, the Sakoe-Chiba and Polynomial constraint both produced a positive result, since the path is still in the boundary window and does not leave this path during the entire sequence. However, the order of the polynomial could change the result.

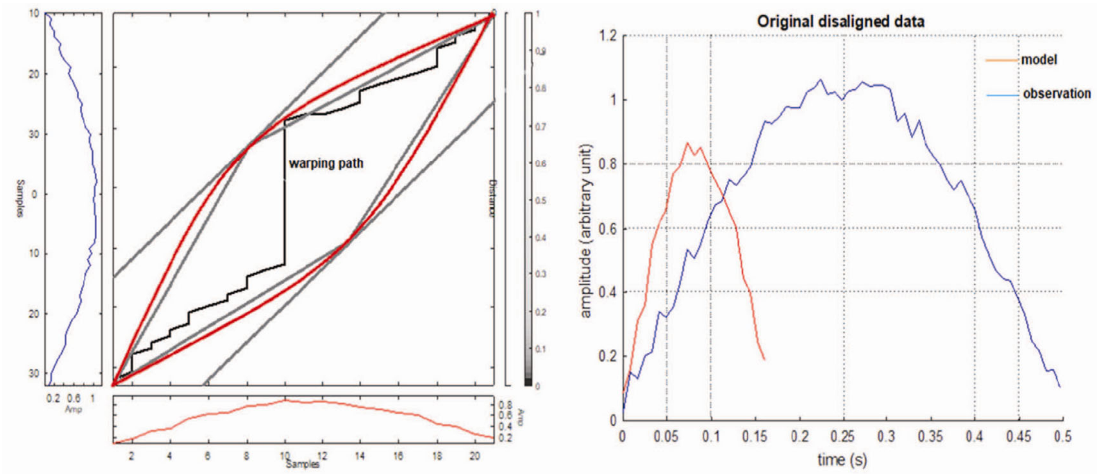


Fig. 14. Polynomial constraint performance evaluation

Further investigations of the methods, in which some orders of the polynomial constraint were also considered and subjected to a coordinate transformation, are shown below. A coordinate transformation facilitates understanding and provides clarity. The coordinate system was only rotated by 45 degrees in a mathematically counterclockwise direction. Fig. 15 illustrates the similarity search of five time series using three constraints.

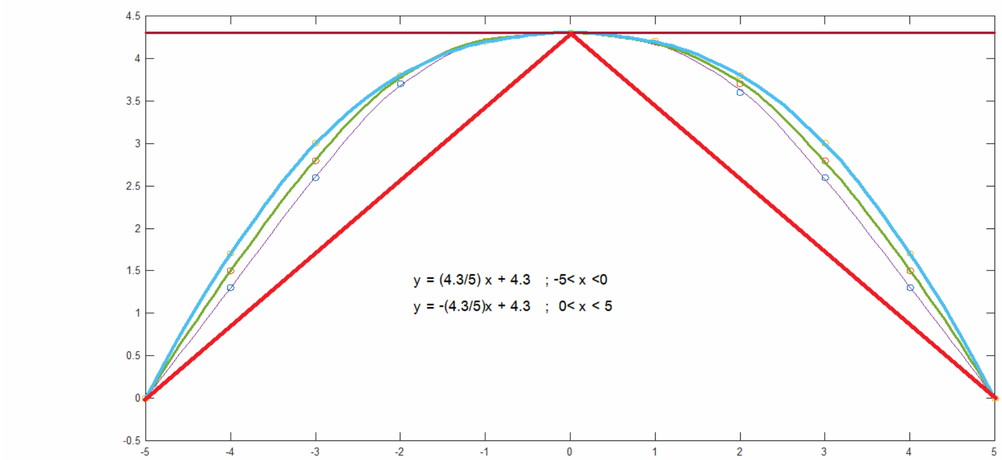


Fig. 15. Example 1- similarity search of five time series

The Itakura algorithm of Fig. 15 has evaluated that the red warping path is most similar to the model. The remaining paths would be discarded by the same algorithm, although some of them are more similar to the model. This becomes clear when you look at the units. The red warping path when compared to the remaining four time series, has the worst resemblance with 60.00570 units to the model. It can be seen that in fact the turquoise path with 46.91766 units shows the best resemblance to the model. By exiting the window, the polynomial constraint would have accepted this path.

The next best similarity can be seen in the pink warping path with 49.23164 unit. This time series has the second-best resemblance to the model. This is not discarded with the polynomial constraint. The remaining three paths would not be discarded at a certain degree of the polynomial method and could be used for further processing to get the best possible similarity. In this example, the efficiency of the polynomial constraint in comparison to the Itakura constraint becomes clear.

From the evaluation of Fig. 16, it can be seen that according to the Itakura algorithm, the red warping path has the best similarity to the model while the remaining four time series would be excluded from the classification. This warping path would continue to be rejected by any presented bounding window. A look at the units of the individual time series shows that the purple warping path with 43.70253 unit actually has the best resemblance to the model. This path would be discarded by the Itakura algorithm and not further considered for classification. On the other hand, the polynomial constraint is again advantageous. After a certain choice of polynomial degree, all five time series are not excluded from the classification. Thus, the best time series that gives the purple warping path can be predicted in the classification as the most similar time series to the model among the remaining four time series.

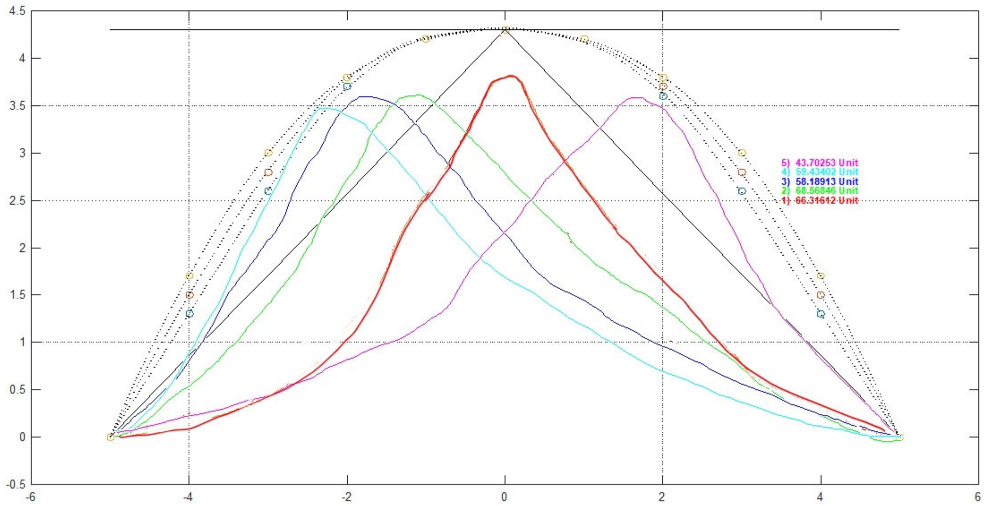


Fig. 16. Example 2- similarity search of five time series

In this example of Fig. 17, all the time series except the one belonging to the red path is again excluded from the Itakura algorithm. A look at the units shows us that the purple warping path with 31.33688 unit provides the best resemblance to the model. This path would not be excluded by polynomial constraint. This algorithm thus provides better flexibility and more effective classification when compared to the Itakura algorithm.

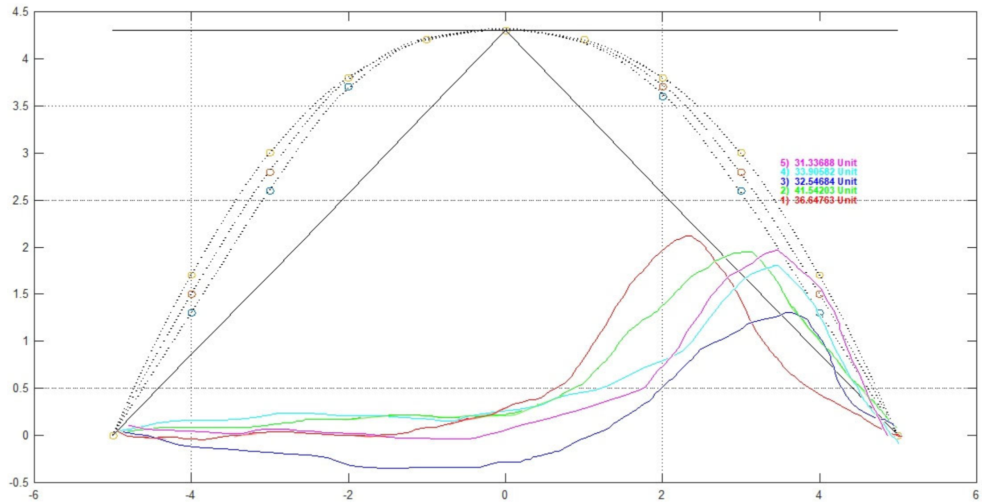


Fig. 17. Example 3- similarity search of five time series

In summary, although the Sakoe - Chiba method does not present a problem in any of the three examples but it does not consider the boundary conditions in its classification. This poses a major problem for qualitative and effective classification. Moreover, it is clear from the examples that in comparison between Itakura and polynomial constraint, the latter is capable of providing much better predictions which can be achieved by a simple, smart choice of the polynomial.

The polynomial boundary has been evaluated and illustrated in MATLAB. Furthermore, previously discussed window functions were compared and examined for their quality and effectiveness. Fig. 18 illustrates the similarity search of a single time series using three (Sakoe - Chiba, Itakura and Polynomial) constraints. The yellow area in the figure represents the space between Itakura and Polynomial (red) constraints, where more flexibility is granted to the warping path. Following that, Fig. 19 demonstrates results for a similarity search conducted over multiple time series as comparison.

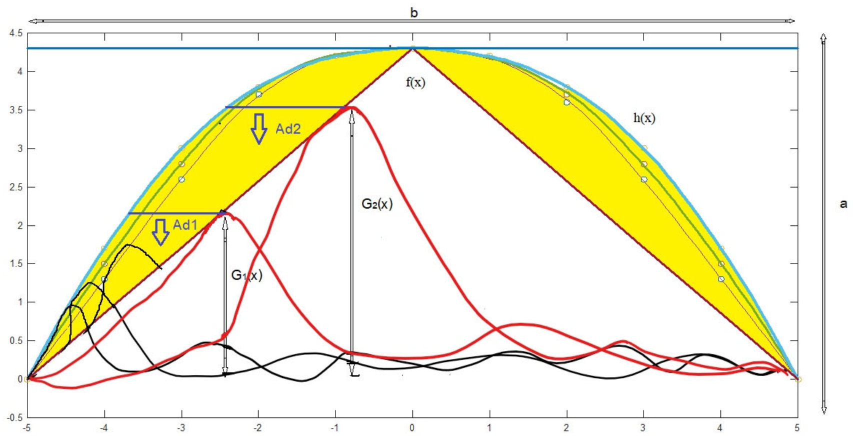


Fig. 18. Similarity search of a single time series

The area A_d between Itakura f(x) and Polynomial h(x) functions can be calculated as follows:

$$f(x) = s \cdot x + p \begin{cases} s = \dfrac{2a}{b},\ p = b \\ s = \dfrac{2a}{b},\ p = b \end{cases} \quad \textbf{(12)}$$

$$h(x) = a \cdot x^8 + b \cdot x^7 + c \cdot x^6 + d \cdot x^5 + e \cdot x^4_{x^8} + f \cdot x^3 + g \cdot x^2 + h \cdot x + i \quad \textbf{(13)}$$

$$A_d = \int_{-\frac{b}{2}}^{\frac{b}{2}} (h(x) - f(x))dx$$
$$with\ x \in \left[-\frac{b}{2}, \frac{b}{2}\right] \quad \textbf{(14)}$$

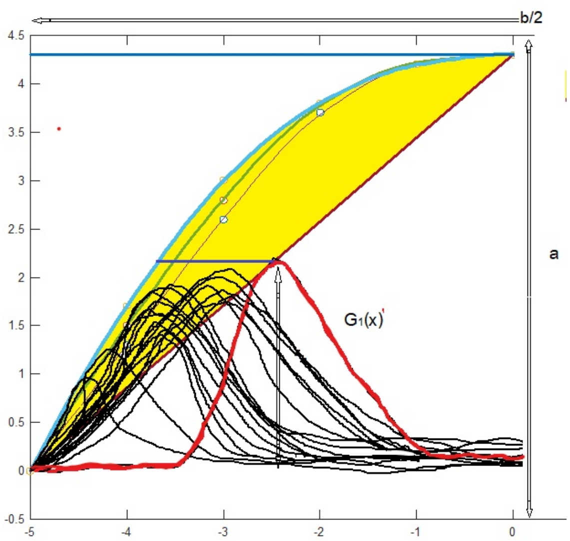


Fig. 19. Similarity search of a multiple time series

One of the main issues with this method, however, was figuring out how to fill the entire area with the mathematic representation. First, once the continuous line is drawn uninterrupted from the beginning to the end while maintaining the same curvature as the red one but smaller in the peak. Then, if the line drawing is implemented by keeping closed contact with one another during the entire length of the drawing, then the lines are seen to be layered, i.e. stacked one on top of the other, all sharing the same outer edges.

Assuming the shortest arbitrary warping path under Itakura constraints presents us with the function $G_i(x)$ and the generated area A_{di}.

From equation (12), where, *a* is the size of the warping window *b* is the length of the diagonals, the number of all possible paths which generated the area represented in the summation of area as A_{sd} in the space between Itakura f(x) and Polynomial h(x) methods results (highlighted in yellow in Fig. 18):

$$A_{sd} = 2 \cdot \sum_{i=0}^{a} \left(\sum_{x=\frac{b}{2}}^{G_j(x)_{\max}} [h(x_i) - f(x_i)] \cdot \Delta x_i \right) \cdot i \qquad \text{with } x \in \left[-\frac{b}{2}, 0 \right] \tag{15}$$

From equation (15), where, $G_1(x)_{max}$ is the maximum value of the first shortest path under the Itakura constraint.

A_{sd} mathematically represents the maximum number of shortest discarded paths under Itakura constraint.

4 Implementation and Results

Our implementation is conducted on CPU Intel(R) Core (™) i7 @2.93GHz with 8GB RAM and NVidia GeForce GTX 750 Ti 2GB GDDR5 RAM with CUDA version 8.0 (Compute Capability 5.0). CUDA-enabled Graphic Processing Units (GPUs) are high-computing-power, cost-effective and power-efficient mutlicore architectures appropriate for accelerating the execution of a wide range of algorithms [16]. Our methods are implemented in a C++/CUDA framework on Windows. An example such implementation of subsequence search with DTW for the CUDA-programming model and FPGAs was given by Sart et al. in [17]. From the CUDA performance guidelines [18], it recommend to use of the shared memory for data reuse and data sharing. Astonishingly, these recommendations were reviewed and implemented by Volkov [19-20], who stated that an extensive use of the shared memory can deliver the optimal performance. The speedups achieved by are approximately from 10 to 14 times better than a CPU implementation using MMX and SSE extensions [21].

First of all, we needed to find the best model as a query to be normalized. We had 23 datasets of data collected before the previous earthquakes. They are obtained from earthquaketrack.com in the range of 350kms from Chiang Mai station (station IU CHTO) within 4 years. Our method to find the model is to apply Dynamic Time Warping (DTW) assuming the first dataset from 23 earthquakes as a query and search through the rest of the 22 datasets. We applied this method to every dataset, selected at least two error sets, and merged them.

After we got our model dataset, we implemented z-normalization to it. Then, to evaluate, we used this model and applied DTW with 3 types of constraints- polynomial constraint, Sakoe-Chiba constraint, and Itakura constraint. We mainly focus on search time in comparison between polynomial constraint and Sakoe-Chiba constraint as well as on the percentage of dismissing found in comparison between polynomial constraint and Itakura constraint. Moreover, we have chosen to use 3 sizes of query (number of data) to evaluate the search time.

The Parameters of polynomial constraint in Table 1 are based on the coefficients of the following formula:

$$ax^8 + bx^7 + cx^6 + dx^5 + ex^4 + fx^3 + gx^2 + hx + i \tag{16}$$

For instance, in the first row of Table 1, the parameters of polynomial constraint are -0.0001, 0.0001, 0.0010, -0.0001, -0.0159, 0.0031, -0.1161 -0.0237, and 4.3169. Those parameters represent *a, b, c, d, e, f, g, h* and *i* respectively.

Table 1. Comparison of Polynomial constraint, Sakoe-Chiba constraint, and Itakura constraint

Size of query	Parameters of polynomial constraint	Search Time (ms)		Dismissing paths compared with Itakura (%)
		Polynomial	Sakoe-Chiba	
1000	-0.0001 0.0001 0.0010 -0.0001 -0.0159 0.0031 -0.1161 -0.0237 4.3169	22,341	26,455	8.69
1000	-0.0001 0.0001 0.0008 -0.0001 -0.0143 0.0031 -0.0924 -0.0237 4.3033	23,881	26,152	8.69
1000	-0.0001 0.0001 0.0001 -0.0001 -0.0056 0.0001 -0.1055 -0.0001 4.3060	25,013	26,597	13.04
2000	-0.0001 0.0001 0.0010 -0.0001 -0.0159 0.0031 -0.1161 -0.0237 4.3169	18,132	22.309	8.69
2000	-0.0001 0.0001 0.0008 -0.0001 -0.0143 0.0031 -0.0924 -0.0237 4.3033	18,368	22.148	8.69
2000	-0.0001 0.0001 0.0001 -0.0001 -0.0056 0.0001 -0.1055 -0.0001 4.3060	19,020	21,951	13.04
4000	-0.0001 0.0001 0.0010 -0.0001 -0.0159 0.0031 -0.1161 -0.0237 4.3169	13,251	15,872	8.69
4000	-0.0001 0.0001 0.0008 -0.0001 -0.0143 0.0031 -0.0924 -0.0237 4.3033	14,277	15,881	8.69
000	-0.0001 0.0001 0.0001 -0.0001 -0.0056 0.0001 -0.1055 -0.0001 4.3060	14,447	15,749	13.04

From Table 1, the comparison between polynomial constraint and Sakoe-Chiba constraint is conducted with different parameters of polynomial coefficients and the window size is the same as Itakura band which is 53. Window size is the length between two boundary lines to limit the scope of warping paths (variable a in Fig. 13).

The polynomial parameter determines the boundary of the searching path and affects the computation time and also reduces the dismissing path depending on the size of the data query. To find the best polynomial parameter, we still need an optimum way to adapt it to the size of data query.

4 Conclusion

In conclusion, two time series can have the same meaning, but the time lengths do not have to be the same. Many classification algorithms would see little similarities in such two time series. The individual elements of the time series would be subject to a direct comparison. Therefore, a time normalization is required in which the time series are stretched or compressed. However, this must not simply be proportional, since the variation in time does not affect all the time series parts equally. A standardization procedure that takes this fact into account is Dynamic Time Warping (DTW). Both the observation and the model correspond to two realizations of the same time series. The non-linear time distortion was detected there with DTW. These are the connecting lines between observation and model. They have been dynamically adapted. If the time series are displayed in the respective axes of the coordinate system, certain methods can be used to calculate the error size.

In this paper, we have introduced a polynomial constraint technique for DTW with time series seismic data pattern recognition in order to predict future earthquakes. We have presented comparisons of our constraint with state-of-the-art constraints- Sakoe-Chiba constraint and Itakura constraint. Our experiments have an estimated accuracy of 0.95. The implementation of polynomial constraint methodology is proved and used to reduce to unattended rejection warping paths, thereby, improving accuracy over the other two constraints.

Similar constraint can also be applied in many other projects, such as- Pattern Recognition for Large-scale and Incremental Time Series in Healthcare, Risky Driver Recognition Based on Vehicle Speed Time Series, Searching for a Similar Time Series Fragment when Solving a Recognition Problem via the Example of an Electrocardiogram, Symbolic representations of time series applied to biometric recognition based on ECG Signals and so on. Time series is also applied in many areas such as financial [22], sensor networks [23] and energy industry [24], but It is important to note that the landslide evolution process is often completely nonlinear and chaotic [25-26]. Then we also supposed to develop the similarity model and techniques [27-28] with noise tolerance and translation.

Acknowledgements

This research was done in collaboration with the Brownien Laboratory of CAT Telecom PCL, Thailand, Chulalongkorn University and Sripatum University. We would like to give special thanks to the lab support team for giving us the idea and motivation to continue this endeavor and for supporting us on this research. This research also required collection of vast amount of data which had to be created from scratch and properly classified. For that, we would like to acknowledge and thank our colleagues for their utmost support and hard work of data collection and arrangement of the said data into the necessary datasets required for the training and testing this model.

References

[1] P. Esling, C. Agon, Time-series data mining, Acm Computing Surveys 45(1)(2012) 7-7.

[2] Furuta, Y. Nomura, Time series prediction of earthquake input by using soft computing, in: Proc. Fourth International Symposium on Uncertainty Modeling and Analysis, 2003.

[3] N. Rajabi, O. Rajabi, Real time earthquake prediction using cross-correlation analysis & transfer function model, in: Proc. 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2015.

[4] C. Li, X. Liu, An improved PSO-BP neural network and its application to earthquake prediction, in: Proc. 2016 Chinese Control and Decision Conference (CCDC), 2016.

[5] Y. Lou, H. Ao, Y. Dong, Improvement of dynamic time warping (DTW) algorithm, in: Proc. 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), 2015.

[6] V. Tuzcu, S. Nas, Dynamic time warping as a novel tool in pattern recognition of ECG changes in heart rhythm disturbances, in: Proc. 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005.

[7] S. Fong, Using hierarchical time series clustering algorithm and wavelet classifier for biometric voice classification, Journal of Biomedicine and Biotechnology 2012(2012). doi: 10.1155/2012/215019.

[8] P. Senin, Dynamic Time Warping Algorithm Review, Unpublished.

[9] SEED, Standard for the Exchange of Earthquake Data: Reference Manual, Incorporated Research Institutions for Seismology, Arlington, 1990.

[10] SAX-VSM, Z-normalization of time series. <https://jmotif.github.io/sax-vsm_site/morea/algorithm/znorm.html>, 2016.

[11] A. Mueen, E. Keogh, Extracting optimal performance from dynamic time warping, in: Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD, 2016.

[12] G.E. Batista, X. Wang, E.J. Keogh, A complexity-invariant distance measure for time series, in: Proc. the 2011 SIAM International Conference on Data Mining, 2011.

[13] G. Mantena, S. Achanta, K. Prahallad, Query-by-example spoken term detection using frequency domain linear prediction and non-segmental dynamic time warping, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22(5)(2014) 946-955.

[14] S. Salvador, P. Chan, FastDTW: Toward Accurate Dynamic Time Warping in Linear Time Space, Florida Institute of Technology, Melbourne, 2004.

[15] C. Fang, From Dynamic time warping (DTW) to Hidden Markov Model (HMM), University of Cincinnati, 2009.

[16] E. Lindholm, J. Nickolls, S. Oberman, J. Montrym. NVIDIA Tesla: a unified graphics and computing architecture, IEEE Micro 28(2)(2008) 39-55.

[17] D. Sart, A. Mueen, W. Najjar, E. Keogh, V. Niennattrakul, Accelerating dynamic time warping subsequence search with GPUs and FPGAs, in: Proc. 2010 IEEE International Conference on Data Mining, 2010.

[18] CUDA, C Programming guide, Tech. Rep., Feb. 2014.

[19] V. Volkov, J. Demmel, Benchmarking GPUs to tune dense linear algebra, in: Proc. International Conference for High Performance Computing, Networking, Storage and Analysis, 2008.

[20] V. Volkov, Better performance at lower occupancy, in: Proc. the GPU Technology Conference, 2010.

[21] W.J. van der Laan, A.C. Jalba, J.B. Roerdink, Accelerating wavelet lifting on graphics hardware using CUDA, IEEE Trans. Parallel Distrib. Syst. 22(1)(2011) 132-146.

[22] D. Reid, A.J. Hussain, H. Tawfik, Financial time series prediction using spiking neural networks, PLoS ONE 9(8)(2014).

[23] J. Pardo, F. Zamora-Martínez, P. Botella-Rocamora, Online learning algorithm for time series forecasting suitable for low cost wireless sensor networks nodes, Sensors 15(4)(2015) 9277-9304.

[24] J. S. Bojanowski, A. Vrieling, A. K. Skidmore, A comparison of data sources for creating a long-term time series of daily gridded solar radiation for Europe, Solar Energy 99(2014) 152-171.

[25] S.Q. Qin, J.J. Jiao, S.J. Wang, The predictable time scale of landslides, Bulletin of Engineering Geology and the Environment 59(4)(2001) 307-312.

[26] S. Qin, J.J. Jiao, S. Wang, A nonlinear dynamical model of landslide evolution, Geomorphology 43(1-2)(2002) 77-85, 2002.

[27] C. Faloutsos, R. Agrawal, A. Swami, Efficient similarity search in sequence databases, in: Proc. the Fourth Int. Conf. on Foundations of Data Organizations and Algorithms, 1993.

[28] H.S. Sawhney, R. Agrawal, K. Lin, K. Shim, Fast similarity search in the presence of noise, scaling and translation in time-series databases, in Proc. 21st VLDB Conference, 1995.