TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH
TẠP CHÍ KHOA HỌC
KHOA HỌC TỰ NHIÊN VÀ CÔNG NGHỆ
Tập 15, Số 3 (2018): 148-160
ISSN: 1859-3100

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE
NATURAL SCIENCES AND TECHNOLOGY
Vol. 15, No. 3 (2018): 148-160

Email: tapchikhoahoc@hcmue.edu.vn; Website: http://tckh.hcmue.edu.vn

PATTERN MATCHING UNDER DYNAMIC TIME WARPING FOR TIME SERIES PREDICTION

Nguyen Thanh Son*
Faculty of Information Technology Ho Chi Minh City University of Technology and Education
Received: 01/11/2017; Revised: 11/12/2017; Accepted: 26/3/2018

ABSTRACT

Time series forecasting based on pattern matching has received a lot of interest in the recent years due to its simplicity and the ability to predict complex nonlinear behavior. In this paper, we investigate into the predictive potential of the method using k-NN algorithm based on R-tree under dynamic time warping (DTW) measure. The experimental results on four real datasets showed that this approach could produce promising results in terms of prediction accuracy on time series forecasting when comparing to the similar method under Euclidean distance.*

Keywords: dynamic time warping, k-nearest neighbor, pattern matching, time series prediction.

TÓM TẮT

Dự báo trên chuỗi thời gian bằng phương pháp so trùng mẫu dưới độ đo xoắn thời gian động

Dự báo trên chuỗi thời gian đã và đang nhận được nhiều quan tâm nghiên cứu trong những năm qua do tính đơn giản và khả năng dự báo trên các chuỗi thời gian phi tuyến phức tạp. Trong bài báo này, chúng tôi nghiên cứu sử dụng thuật toán k-NN dựa trên R-tree dưới độ đo DTW cho bài toán dự báo trên chuỗi thời gian. Các kết quả thực nghiệm trên bốn tập dữ liệu thực cho thấy cách tiếp cận này có thể cho kết quả dự báo chính xác hơn khi so sánh với phương pháp tương tự sử dụng độ đo Euclid.*

Từ khóa: dự báo trên chuỗi thời gian, k lân cận gần nhất, so trùng mẫu, xoắn thời gian động.

1. Introduction

A time series is a sequence of real numbers where each number represents a value at a given point in time. Time series data arise in so many applications of various areas ranging from science, engineering, business, finance, economy, medicine to government.

An important research area in time series data mining which has received an increasing amount of attention lately is the problem of prediction in time series. A time series prediction system predicts future values of time series variables by looking at the collected variables in the past. The accuracy of time series prediction is fundamental to many decision processes and hence the research for improving the effectiveness of prediction methods has never stopped.

* Email: *sonnt@fit.hcmute.edu.vn*

One thing the pattern matching-based forecasting has in common is it needs to find the best match to a pattern from a pool of time series in the past. The Euclidean distance metric has been widely used for pattern matching [1]. However, its weakness is sensitive to distortion in time axis [2]. For example, in the case of the pattern and a candidate time series have an overall similar shape but they are not aligned in the time axis, Euclidean distance will produce a pessimistic dissimilarity measure but the DTW distance can produce a more intuitive distance measure. Figure 1 illustrates this case.

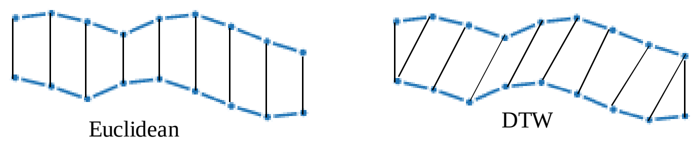


***Figure 1.** An example illustrates the Euclidean distance and the DTW distance*

In our work, we investigate into the predictive potential of the DTW-based pattern matching technique on time series and compare it to the similar method under Euclidean distance. The pattern matching method here is the *k*-nearest neighbor method. The *k*-nearest neighbor algorithm is selected because it is simple and it can work very fast.

The DTW-based pattern matching technique for time series prediction performs as follows: first, it retrieves the pattern (subsequence) prior to the interval to be forecasted. Then this pattern is used for searching *k* nearest neighbors under DTW distance measure in history data. Next, subsequences next to these found *k* nearest neighbors are retrieved. Finally, the forecasted sequence is calculated by averaging the subsequences found in the immediate previous step.

The dynamic time warping distance measure is used because it is introduced as a solution to the weakness of Euclidean distance metric [3].

The experimental results on four real datasets showed that this approach can produce promising results on time series in comparison with forecasting method using *k*-NN algorithm under Euclidean distance measure.

The rest of the paper is organized as follows. Section 2 examines background and related words. Section 3 describes our approach for forecasting in time series. Section 4 presents our experimental evaluation on real datasets. In section 5 we include some conclusions.

2. Background and related works

2.1. Background

- ***Euclidean Distance***

Euclidean distance is the simplest method to measure the similarity of time series. Given two time series $Q = \{q_1, \ldots, q_n\}$ and $C = \{c_1, \ldots, c_n\}$, the Euclidean distance between Q and C is defined as

$$D(Q,C) = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \quad (2.1)$$

- ***Dynamic time warping distance.***

In 1994, the DTW technique is introduced to the database community by Berndt and Clifford [3]. This technique allows similar shapes to match even if they are out of phase in the time axis. So, it is widely used in various fields such as bioinformatics, chemical engineering, robotics, and so on.

Given two time series Q of length n, $Q = \{q_1, \ldots, q_n\}$, and C of length m, $C = \{c_1, \ldots, c_m\}$, the DTW distance between Q and C is calculated as follows.

First, an n-by-m matrix is constructed where the value of the (i^{th}, j^{th}) element of the matrix is the squared distance $d(q_i, c_j) = (q_i - c_j)^2$. To find the best distance between the two sequences Q and C, a path through the matrix that minimizes the total cumulative distance between them is retrieved. A warping path, $W = w_1, w_2, \ldots, w_L$ with $\max(m, n) \le L \le m+n-1$, is an adjacent set of matrix elements that defines a mapping between Q and C. The optimal warping path is the path which has the minimum warping cost. It is defined as.

$$DTW(Q,C) = \min_{W}\left\{\sum_{k=1}^{L} d_k, W = \langle w_1, w_2, \ldots, w_L \rangle\right\} \quad (2.2)$$

where $d_k = d(q_i, c_j)$ indicates the distance represented as $w_k = (i, j)_k$ on the path W.

To find the warping path, we can use dynamic programming which is calculated by the following formula.

$$\gamma(i, j) = d(q_i, c_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \quad (2.3)$$

where $d(q_i, c_j)$ is the distance found in the current cell, $\gamma(i, j)$ is the cumulative distance of $d(i, j)$ and the minimum cumulative distances from the three adjacent cells.

Figure 2 shows an example of how to calculate the DTW distance between two time series Q and C.

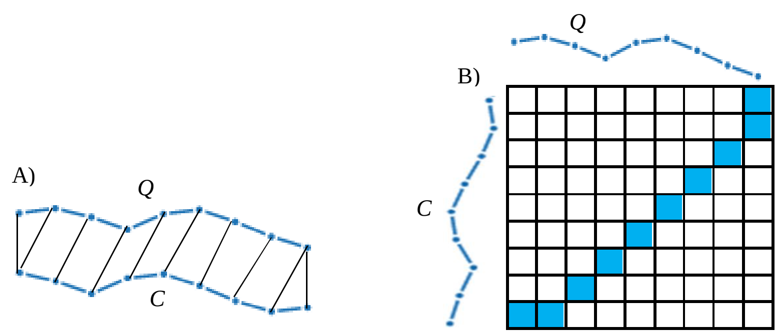


Figure 2. *An example of how to calculate the DTW distance between Q and C. (A) Two similar but out of phase time series Q and C. (B) To align two time series, a warping matrix is constructed for searching the optimal warping path.*

A recent improvement of DTW that considerably speeds up the DTW calculation is a lower bounding technique based on the warping window [2]. Figure 3 illustrates the Sakoe-Chiba Band [4] and the Itakura Parallelogram [5] which are two most common constraints in the literature.

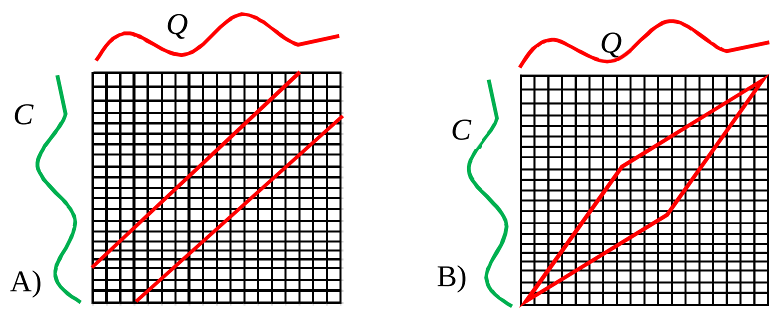


***Figure 3.** An example illustrates (A) Sakoe-Chiba Band and (B) Itakura Parallelogram*

According to this technique, sequences must have the same length. If the sequences are of different lengths, one of them must be re-interpolated. In order to enhance the search performance in large databases, first a warping window is used to create an above bounding line and a below bounding line (called bounding envelope) of the query sequence. Then the lower bound is calculated as the squared sum of the distances from every part of the candidate sequence not falling within the bounding envelope, to the nearest orthogonal edge of the bounding envelope. Figure 4 illustrates this technique.

The complexity of DTW algorithm using dynamic programming is O(*nm*), where *n* and *m* are the length of sequences [2]. However, in [2], Keogh and Ratanamahatana proposed a linear-time lower bounding functions to prune away the quadratic-time computation of the full DTW algorithm.

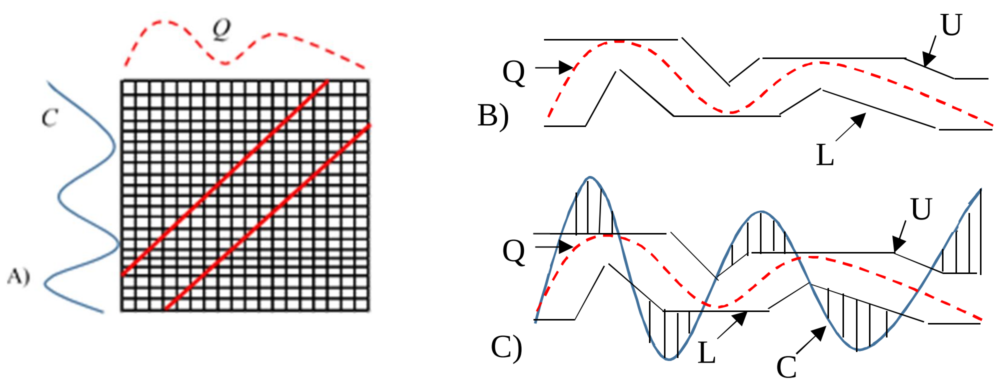


***Figure 4.** (A) The Sakoe-Chiba Band is used to create a bounding envelope. (B) The bounding envelope of a query sequence Q. (C) The lower bound for DTW distance retrieved by calculating the Euclidean distance between any candidate sequence C and the closest external part of the envelope around a query sequence Q.*

2.2. Related works

Various kinds of prediction methods have been developed by many researchers and business practitioners. Some of the popular methods for time series prediction, such as exponential smoothing ([6]), ARIMA model ([7], [8], [9]), artificial neural networks (ANNs) ([10], [11], [12], [13], [14], [15]) and Support Vector Machines (SVMs) ([16], [17]) are successful in some given experimental circumstances. For example, the exponential smoothing method and ARIMA model are linear models and thus they can only capture the linear features of time series. ANN has shown its nonlinear modeling capability in time series forecasting, however, this model is not able to capture seasonal or trend variations effectively with the un-preprocessed raw data [15].

Some pattern matching methods are also introduced for time series prediction such as:

In 2009, Arroyo and Mate proposed a time series forecasting method which adapts *k*-nearest neighbor method to forecasting histogram time series (HTS) [18]. This HTS is used to describe situations where a distribution of values is available for each instant of time. The authors showed that this method can yield promising results.

In 2013, Zhang et al. presented a *k*-nearest neighbor model for short-term traffic flow prediction [19]. First, this method preprocesses the original data and then standardizes the processed data in order to avoid the magnitude difference of the sample data and improve the prediction accuracy. At last, a short-term traffic prediction based on *k*-NN nonparametric regression model is carried out.

In 2015, Cai et al. proposed an improvement on the *k*-NN model for road speed forecast based on spatiotemporal correlation [20]. This model defines the current conditions by the two-dimensional spatiotemporal state matrices, instead of the one-dimensional state vector of the time series and determines the weights by Gaussian function to adjust the matching distance of the nearest neighbors.

In 2016, Gong et al. proposed a classifier based on UCR Suite and the Support Vector Machine for subsequence pattern matching in financial time series. The result of the classifier are used by financial analysts for predicting price trends in stock markets [21].

Some hybrid methods are also introduced for time series prediction. Some typical methods can be reviewed briefly as follows: Lai et al. (2006) proposed a new hybrid method which combines exponential smoothing and neural network for Financial Time Series Prediction [22]. Truong et al. (2012) proposed a new method which combines motif information and neural network for time series prediction [23]. Bao et al. (2013) introduced a hybrid method which combines Winters' exponential smoothing method and neural network is proposed for forecasting seasonal and trend time series [24]. Also in this year, Son et al. (2013) proposed a hybrid method which is a linear combination of ANN and pattern matching under Euclidean distance-based forecasting method [25]. Mangai et al. (2014) proposed a hybrid method which combines ARIMA model and HyFIS model for

Forecasting Univariate Time Series [26]. Pandhiani and Shabri (2015) introduced a time series forecasting method using hybrid model for Monthly Streamflow Data [27]. This model is developed by integrating an artificial neural network model and least square support vector machine model.

In recent years, a newly emerging area is of Evolving Intelligent systems which can be used for forecasting on data streams. The proposed methods in this direction are online algorithms and usually based on fuzzy rules and evolutionary algorithms. Some methods are introduced in dealing with non-stationary data streams, such as Pratama et al. proposed the scaffolding type-2 classifier for incremental learning under concept drifts [28], the online active learning in data stream regression based on evolving generalized fuzzy models [29], the Incremental Rule Splitting in Generalized Evolving Fuzzy Systems [30].

3. Our proposed approach

Our approach hinges on predicting samples in a time series based on finding its *k* nearest neighbors under the DTW measure. In similarity search, a lower bounding distance measure can help prune sequences that could not be the best match [2]. Besides, a multidimensional index structure (e.g., R-tree or R*-tree) can be used to enhance the search performance in large databases. In this case, a multidimensional index structure can be used for retrieving nearest neighbors of a query.

Figure 5 shows the basic idea of our approach. Our approach for forecasting is described as follows: Given the current state (pattern) of length *w* in the time series that we have to predict a sequence of the next time step. First, the algorithm searches for *k* nearest neighbors under DTW distance. Then the sequences next to the found neighbors are retrieved. Finally, the forecasted sequence is estimated by averaging the sequences found in the immediate previous step. In the case of forecasting more patterns, the estimate sequence is inserted at the end of the data in order to predict the following pattern.

With this approach, the length of prediction can be as long as required because it is implemented with a loop in which forecasting samples can be able to insert in the data set in order to predict further samples. Figure 5 shows the basic idea of our approach.

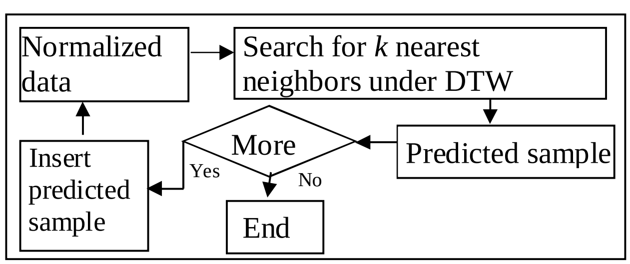


Figure 5. *The basic idea of our approach*

Figure 6 illustrates a *k*-NN algorithm for similarity search problem using a multidimensional index structure which is similar to an algorithm introduced in [2]. In this algorithm, a priority queue is used to contain visited nodes in the index in the increasing order of their distances from query *Q*. The distance defined by $D_{region}(Q, R)$ is used to search in R*-tree. If the current item is a data item, the true distance under DTW(*Q*, *C*) is used. A sequence *C* is moved from *item_list* to *kNN_result* if it is one of the *k* nearest neighbors.

```
Algorithm:  Finding k nearest neighbors using R*-tree
Input: Time series database D, a query Q and k, the number of nearest neighbors
Output: k nearest neighbors
---------------------------------------------------------------
distance = 0
Push root node of index and distance into queue
while queue is not empty
        curr_item = Pop the top item of queue
        if curr_item is a non-leaf node
                for each child node U in curr_item
                        distance = D_region(Q, R)
                        Push U and distance into queue
                end for
        else if curr_item is a leaf node
                for each data item C in curr_item
                        distance = D_region(Q, R)
                        Push C and distance into queue
                end for
        else
                Retrieve original sequence of C from database
                distance = DTW(Q, C)
                Insert C and distance into item_list
        end if
        for each sequence C in item_list which conforms to
    the condition D(Q,C) ≤ curr_item.Distance
            remove C from item_list
            Add C to kNN_result
            If | kNN_result| = k   return kNN_result
    end for
end while
```

Figure 6. *The k-nearest neighbor algorithm for similarity search problem*

Our approach for forecasting is described as follows: Given the current state (pattern) of length *w* in the time series that we have to predict a sequence of the next time step. First, the algorithm searches for *k* nearest neighbors of that pattern under DTW distance. Then the subsequences next to the found neighbors are retrieved. Finally, the forecasted sequence is estimated by averaging the subsequences found in the immediate previous step. In the case of forecasting more patterns, the estimate sequence is inserted at the end of the data in order to predict the following pattern. Figure 7 illustrates the steps of the prediction algorithm based on pattern matching under DTW.

Algorithm: Time series forecasting based on pattern matching under DTW

Input: Time series *D* of length n_1, the length of current pattern *w*, the number of nearest neighbors *k* and the length of predicted sequence *m* ($m \le w << n_1$).

Output: Estimated sequence *S* of length *m*.

1. Reduce the dimensionality of subsequences of length *w* in *D* and insert them into a multidimensional index structure (if necessary).
2. Retrieve the subsequence *S* of length *w* prior to the subsequence we have to predict in *D*.
3. Search for *k* nearest neighbors of *S* under DTW distance.
4. For each nearest neighbor found in step 3, retrieve subsequence of length *m* next to it in *D*.
5. Average subsequences found in step 4.
6. Output the estimated sequence in step 5.
7. Insert the sequence estimated in step 5 into *D* to forecast following pattern and return to step 1 (if necessary).

Figure 7. *The algorithm for prediction based on pattern matching using DTW distance*

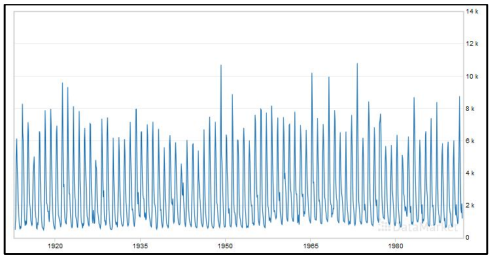
Note that, in the case of $m < w$ we can use a variable to accumulate the estimated sequences until *m* is equal to *w*. At that time we can insert the accumulated sequence into the used index structure without need to rebuild the whole index structure in step 1.

4. Experimental evaluation.

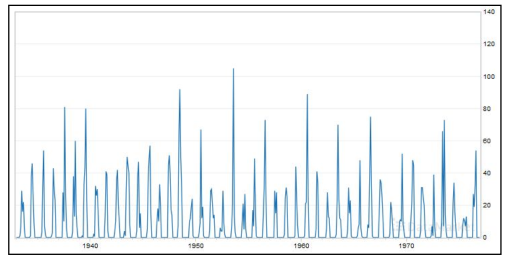
- ***The datasets***

We experiment on four real datasets: Fraser river (FR), Monthly rain (MR), Natural gas (NG), and Stock index (SI). Figure 8 shows the plots of the above datasets. We compare the performance of this prediction approach with that of the forecasting method using *k*-NN algorithm under Euclidean distance measure. We use patterns of length 12, predicted sequences of length 1 and for each experimental dataset we test with some *k* values for *k*-nearest-neighbor search then choose the best one. The length of predicted sequences is 1 since only one-step prediction is considered in this study.

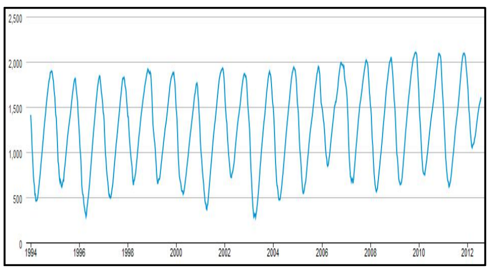
We compare the performance of the two prediction methods on all segments of the test dataset and calculate the mean of errors in the predictive duration. We implemented our method with Microsoft Visual C# and conducted the experiments on a Core i3, Ram 2GB.



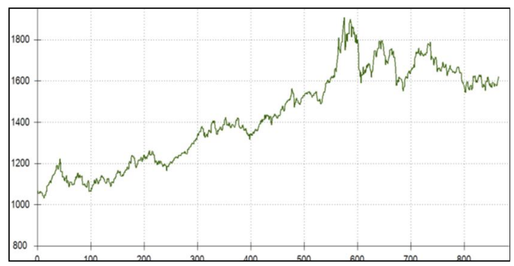
(a) Fraser river



(b) Monthly rain



(c) Natural gas



(d) Stock index

Figure 8. *The four different datasets*

The datasets for experiment are described as follows.

– Fraser river dataset, from 1/1913 to 12/1990 (http://datamarket.com/data/set/22nm/#!display=line&ds=22nm).

– Monthly rain, from 1/1933 to 12/1976 (http://datamarket.com/data/set/22n8/#!display=line&ds=22n8).

– Weekly Eastern Consuming Region Natural Gas Working Underground Storage (Billion Cubic Feet), from the week 31/12/1993 to 27/7/2012 (http://tonto.eia.gov/dnav/ng/hist/nw_epg0_sao_r88_bcfw.htm).

– Stock index S&P 500, from 03/01/2007 to 31/12/2012 (http://www.forexpros.com/indices/usspx- 500-historical-data).

- ***Evaluation criteria***

In this study we use the mean absolute error (MAE), the root-mean-square error (RMSE) and the coefficient of variation of the RMSE, called CV(RMSE) to measure the prediction accuracy. They are defined as follows.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_{obs,i} - Y_{\mathrm{model},i}\right| \quad (3.1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Y_{obs,i} - Y_{model,i})^2}{n}} \quad (3.2)$$

$$CV(RMSE) = \frac{RMSE}{\overline{Y_{obs}}} \quad (3.3)$$

Where Y_{obs} is observed values and $Y_{model,i}$ is modeled value at time i.

- ***Experimental evaluation results***

To examine the impact of *k* on the predictive accuracy, we test with some *k* values. Then averaging the predictive errors. Table 1 shows the predictive mean absolute error (MAE) of the experiment on the monthly rain dataset with *k* from 1 to 10. The experimental result shows that the predictive errors will be changed with different values of *k*. In this experiment we see that the predictive error are minimum if the chosen *k* is 9.

Table 1. *The predictive errors of the experiment on monthly rain dataset with k from 1 to 10*

k	MAE	*k*	MAE
1	0.07917	6	0.07874
2	0.08859	7	0.07962
3	0.08274	8	0.07778
4	0.08477	9	0.07736
5	0.08254	10	0.07798

Table 2 shows the experimental result from the monthly rain dataset with the best *k*. The prediction errors are calculated for each of the last four years. At the end of the table is the mean of error in four years. For brevity, in table 3 we only show the summary of results obtained from the experiment on the four datasets. The values in this table are the means of error in years forecasted.

The experimental results on the above real datasets show that the means of prediction errors in predicted years of the approach under DTW are better than those of the forecasting method using *k*-NN algorithm under Euclidean distance. It means that the prediction method based on pattern matching under pattern matching could produce a prediction result better than that of the pattern matching-based prediction method under Euclidean distance in terms of accuracy.

Table 2. *Experimental result from the monthly rain dataset*

Year	MAE		RMSE		CV(RMSE)	
	k-NN (Euclid)	k-NN (DTW)	k-NN (Euclid)	k-NN (DTW)	k-NN (Euclid)	k-NN (DTW)
1	0.12187	0.11578	0.23065	0.21728	1.65123	1.55550
2	0.04012	0.05265	0.07325	0.08908	1.18331	1.43904
3	0.07619	0.07434	0.14771	0.13570	3.72229	3.41969
4	0.07125	0.06420	0.13727	0.11593	1.31034	1.10663
Mean	0.07736	0.07674	0.14722	0.13950	1.96679	1.88021

Table 3. *The summary of results obtained from the experiment on four datasets*

Dataset	**MAE**		**RMSE**		**CV(RMSE)**	
	k-NN (Euclid)	**k-NN (DTW)**	**k-NN (Euclid)**	**k-NN (DTW)**	**k-NN (Euclid)**	**k-NN (DTW)**
MR	0.07736	0.07674	0.14722	0.13950	1.96679	1.88021
FR	0.04587	0.04586	0.06019	0.06052	0.29474	0.29290
NG	0.05892	0.05484	0.07637	0.06818	0.12878	0.11519
SI	0.01778	0.01681	0.02225	0.02106	0.02795	0.02646

Besides prediction accuracy, we also compare the two methods in terms of prediction (processing) time. Table 4 shows the running time (in seconds) of the two methods over the four datasets. We can see that the running time of the method under DTW is greater than that of the pattern matching-based prediction method under Euclidean distance.

Table 4. *The running time of the two methods on four different datasets*

Dataset	**Runtime (seconds)**	
	DTW-based method	**Euclid-based method**
FR	0.6466	0.1992
MR	0.4325	0.2853
NG	0.2783	0.0984
SI	1.1056	0.7164

5. Conclusions.

In this paper, we have examined the prediction method based on pattern matching using DTW distance for general-purpose time series which have trend and seasonal variations. This approach is compared to the similar method under Euclidean distance in terms of predictive accuracy and processing time.

Our experiments on the above datasets show that the pattern matching-based prediction method under DTW distance could give better prediction accuracy than that of pattern matching-based prediction method under Euclidean distance. However, the running time of the method under DTW is longer than that of the similar method under Euclidean distance.

In future we plan to experiment this method on other datasets and investigate the combination of two measures in time series prediction in order to combine the benefits of these distance measures.

❖ ***Conflict of Interest:*** *Author have no conflict of interest to declare.*

REFERENCES

[1] Keogh and S. Kasetty, "On the Need for Time series Data Mining Benchmarks: A Survey and Empirical Demonstration," *In the 8th ACM SIGKDD*, 2002, pp. 102-111.

[2] E. Keogh, A. Ratanamahatana, "Exact indexing of dynamic time warping," *Journal of Knowledge and Information Systems*, vol. 7 Issue 3, 2005, pp. 358- 386.

[3] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," *AAAI Workshop on Knowledge Discovery in Databases*, 1994, pp. 229-248.

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

[5] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoustics, Speech, and Signal Proc.*, Vol. ASSP-23, 1975, pp. 52-72.

[6] S. Gelper, R. Fried, and C. Croux, "Robust forecasting with exponential and Holt-Winters smoothing," *Journal of Forecasting*, vol. 29, 2010, pp. 285-300.

[7] C. Chatfield. *Time-series forecasting*. New York, NY: Chapman and Hall, Inc, 2000.

[8] I.-B. Kang, "Multi-period forecasting using different models for different horizons: An application to U.S. economic time series data," *International Journal of Forecasting*, vol.19, 2003, pp. 387-400.

[9] J. H. Kim. "Forecasting autoregressive time series with bias corrected parameter estimators," *International Journal of Forecasting*, vol.19, 2003, pp. 493-502.

[10] S. D. Balkin and J. K. Ord, "Automatic neural network modeling for univariate time series," *International Journal of Forecasting*, vol.16, 2000, pp. 509-515.

[11] E. Cadenas and W. Rivera, "Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks," *Renewable Energy*, vol. 34, no. 1, 2009, pp. 274-278.

[12] M. Ghiassi, H. Saidane, and D. K. Zimbra, "A dynamic artificial neural network model for forecasting series events," *International Journal of Forecasting*, vol.21, 2005, pp. 341-362.

[13] S. Heravi, D. R. Osborn and C. R. Birchenhall, "Linear versus neural network forecasting for European industrial production series," *International Journal of Forecasting*, vol.20, 2004, pp. 435-446.

[14] G. Tkacz, "Neural network forecasting of Canadian GDP growth," *International Journal of Forecasting*, vol.17, 2001, pp. 57-69.

[15] G. P. Zhang, M. Qi., "Neural Network Forecasting for Seasonal and Trend Time Series," *European Journal of Operational Research*, vol. 160, 2005, pp. 501-514.

[16] K. J. Kim, "Financial time series forecasting using support vector machines," *Neuro-computing*, vol. 55, 2003, pp. 307-319.

[17] Y.Radhika and M.Shashi, "Atmospheric Temperature Prediction using Support Vector Machines," *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, 2009, pp. 55-58.

[18] J. Arroyo and C. Mate, "Forecasting histogram time series with k-nearest neighbor methods," *International Journal of Forecasting 25*, 2009, pp. 192-207.

[19] L. Zhang, Q. Liu, W. Yang, N. Wei, D. Dong, "An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction," In *Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings* from the 13th COTA International Conference of Transportation Professionals (CICTP2013), vol. 96, 2013, pp. 653-662.

[20] P. Cai, Y. Wang, G. Lu and P. Chen, "An Improved k-Nearest Neighbor Model for Road Speed Forecast Based on Spatiotemporal Correlation," *CICTP 2015*, pp. 342-351.

[21] X. Gong, Y. W. Si, S. Fong, R. P. Biuk-Aghai, "Financial time series pattern matching with extended UCR Suite and Support Vector Machine," In *Expert Systems with Applications: An International Journal*, vol. 55, No. C, 2016, pp. 284-296.

[22] K. Lai, L. Yu, S. Wang, W. Huang. "Hybridizing Exponential Smoothing and Neural Network for Financial Time Series Prediction," *Proceedings of 6th International Conference on Computational Science* (*ICCS'06*), vol. 4, 2006, pp. 493-500.

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

[24] D. N. Bao, N. D. K. Vy and D. T. Anh, "A Hybrid Method for Forecasting Trend and Seasonal Time Series," *In Proc. of 2013 IEEE RIVF International Conference on Information and Communication Technologies*, Hanoi, Vietnam, 10-13 November, 2013, pp. 203-208.

[25] N. T. Son and D. T. Anh, "Hybridizing Pattern Matching and Neural Network for Time Series Prediction," In *Proc. of 2013 World Congress on Information and Communication Technologies (WICT 2013)*, Hanoi, Vietnam, 2013, pp. 19-24.

[26] S. A. Mangai, K. Subramanian, K. Alagarsamy, K. and B. Ravi Sankar, "Hybrid ARIMA-HyFIS Model for Forecasting Univariate Time Series," *International Journal of Computer Applications*, vol. 91, no. 5, 2014, pp. 38-44.

[27] S. M. Pandhiani and A. B.Shabri, "Time Series Forecasting by Using Hybrid Models for Monthly Streamflow Data," *Applied Mathematical Sciences*, vol. 9, no. 57, 2015, pp. 2809-2829.

[28] M. Pratama, J. Lu, E. Lughofer, G. Zhang, S. Anavatti, "Scaffolding type-2 classifier for incremental learning under concept drifts," *Neurocomputing*, 26 May, 2016, vol. 91, pp. 304-329, in press (10.1016/j.neucom.2016.01.049).

[29] E. Lughofer and M. Pratama, "On-line Active Learning in Data Stream Regression employing Evolving Generalized Fuzzy Models with Certainty Sampling," *IEEE Transactions on Fuzzy Systems 2017*, online and in-press, DOI: 10.1109/TFUZZ.2017.2654504.

[30] E. Lughofer, M. Pratama, I. Skrjanc, "Incremental Rule Splitting in Generalized Evolving Fuzzy Systems for Autonomous Drift Compensation," accepted for publication at IEEE Transactions on Fuzzy Systems, 2017, volume: PP, Issue: 99, 2017.