

# An Enhanced Support Vector Machine for Faster Time Series Classification

Thapanan Janyalikit<sup>1</sup>, Phongsakorn Sathianwiryakhun<sup>1</sup>, Haemwaan Sivaraks<sup>2</sup>,  
and Chotirat Ann Ratanamahatana<sup>1</sup>(✉)

<sup>1</sup> Department of Computer Engineering, Chulalongkorn University, Phayathai Rd., Pathumwan,  
Bangkok 10330, Thailand

{Thapanan.Ja, Phongsakorn.Sa}@student.chula.ac.th,  
chotirat.r@chula.ac.th

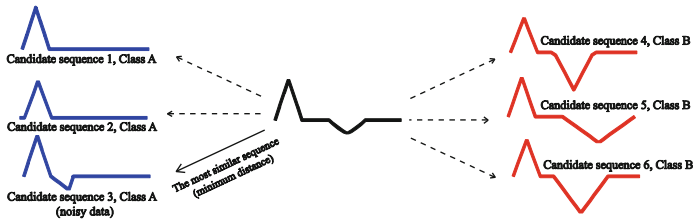
<sup>2</sup> PTT ICT Solution Company Limited, Vidhavadi-Rangsit Rd., Chatuchak,  
Bangkok 10900, Thailand  
haemwaan@gmail.com

**Abstract.** As time series have become a prevalent type of data, various data mining tasks are often performed to extract useful knowledge, including time series classification, a more commonly performed task. Recently, a support vector machine (SVM), one of the most powerful classifiers for general classification tasks, has been applied to time series classification as it has been shown to outperform one nearest neighbor in various domains. Recently, Dynamic Time Warping (DTW) distance measure has been used as a feature for SVM classifier. However, its main drawback is its exceedingly high time complexity in DTW computation, which degrades the SVM's performance. In this paper, we propose an enhanced SVM that utilizes a DTW's fast lower bound function as its feature to mitigate the problem. The experiment results on 47 UCR time series data mining datasets demonstrate that our proposed work can speed up the classification process by a large margin while maintaining high accuracies comparing with the state-of-the-art approach.

**Keywords:** Classification · Support vector machine (SVM) · Dynamic time warping (DTW) · Lower bound · LB\_keogh

## 1 Introduction

Time series classification is widely used in many domains such as medicine, finance, and science [1–4]. One nearest neighbor (1-NN) classifier with Dynamic Time Warping (DTW) distance function is one of the most popular time series classifiers that has been demonstrated to work extremely well and is difficult to beat [5]. Although 1-NN with DTW is quite effective, it still has the known weaknesses in its high computational time and its sensitivities to noises in the training set. Figure 1 illustrates a 2-class problem with six training data sequences, one of which contains noises (candidate sequence 3). When we want to classify a new sequence (with the real class *B* label), the 1-NN with DTW could wrongfully classify it to class *A*, as it matches the best to sequence 3 that is contaminated by noises.



**Fig. 1.** One-nearest neighbor classification with noisy training data

To mitigate this problem, many approaches have been proposed, including Longest Common Subsequence (LCSS) [22] and variation of edit distances [23, 24]. However, they still suffer from high computational cost and difficulty in parameter selection. Some other researchers have proposed Support Vector Machine (SVM) that combined together the benefits of DTW by using DTW distance as the feature [6] to solve time series classification problems [7–9]. SVM is a powerful algorithm that can be applied in many domains because it builds a classifier from a hyperplane that separates features of input or exploits mapping of features into higher dimensional space through kernel function in non-linearly separable problems. It can also ignore some noisy data points that may lead to wrong classification. Recently, some research works have attempted to establish a kernel for DTW [7, 8] or have attempted to use other features together with DTW to increase the accuracy of SVM [9]. However, complexity and computational time for feature construction are still their weaknesses. In this work, we therefore mainly focus on reducing computational time, especially in the feature vector construction step, while being able to maintain high accuracies, comparing with the real DTW distance being used as a feature for SVM. Specifically, we take both the advantages of a lower bound function for DTW and robustness of SVM to build a faster yet accurate time series classifier.

The rest of the paper is organized as follows. The next section gives brief details on background and related works. Section 3 explains our proposed work with details on our experiment setup. Section 4 provides the experiment results and discussion, and the last section concludes our work.

## 2 Background and Related Works

In this section, we briefly give some details about Dynamic Time Warping (DTW), global constraint, Lower bound function for DTW, and Support Vector Machine (SVM).

### 2.1 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is a similarity measure for time series data that has gained its popularity over Euclidean distance metric due to its higher accuracy and more flexibility in non-linear alignments between two data sequences. Given two time series sequences,  $Q$  and  $C$  of length  $m$  and  $n$ , respectively, as follows.

$$Q = \{q_1, q_2, q_3, \dots, q_i, \dots, q_m\} \tag{1}$$

$$C = \{c_1, c_2, c_3, \dots, c_j, \dots, c_n\} \tag{2}$$

To align the two time series, an  $m$  by  $n$  matrix is constructed to hold cumulative distances ( $\gamma(i^{th}, j^{th})$ ) between all pairs of data points from these two time series sequences. The cumulative distance in each cell is calculated from a sum of the distance of the current element,  $d(q_i, c_j) = (q_i - c_j)^2$ , and the minimum of the cumulative distance of the three adjacent elements:

$$\gamma(i, j) = d(q_i, c_j) + \min \{ \gamma(i - 1, j), \gamma(i, j - 1), \gamma(i - 1, j - 1) \} \tag{3}$$

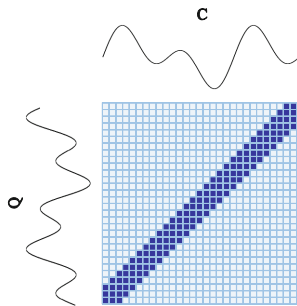
Dynamic programming is used to obtain optimal alignment and cumulative distance from the elements (1, 1) to ( $m, n$ ). The optimal warping path is obtained by backtracking. However, in most time series mining tasks, we often are interested in only the distance value, contained in ( $m, n$ ). So, backtracking can be omitted. The DTW distance is calculated by Eq. (4). Note that the squared root can also be omitted to speed up the calculation.

$$DTW(Q, C) = \sqrt{\gamma(q_m, c_n)} \tag{4}$$

### 2.2 Global Constraint

To improve the performance of DTW, a global constraint is often applied to prevent unreasonable warping, which in turn increases the accuracy. While any shapes and sizes of the global constraints can be utilized, Sakoe-Chiba Band [10] is one of the simplest global constraints [11] that restricts the warping path to stay within specific boundary ( $r$ ), as shown in Fig. 2. Equation (3) with the Sakoe-Chiba constraint can be rewritten as follows:

$$\gamma(i, j) = d(q_i, c_j) + \min \{ \gamma(i - 1, j), \gamma(i, j - 1), \gamma(i - 1, j - 1) \} ; i + r \geq j \text{ and } i - r \leq j \tag{5}$$



**Fig. 2.** The Sakoe-Chiba band restricts the warping path to stay within the boundary.

### 2.3 Lower Bound of DTW

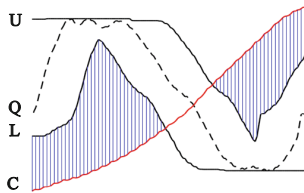
Due to its  $O(n^2)$  time complexity, DTW is considered too expensive for large time series data. In the past decade, various lower bound functions for DTW have been proposed to speed up the DTW calculation for data mining tasks [12–14]. The main idea of lower bound function is to approximate the real DTW distance with the crucial property that the computed lower bound will never exceed the real DTW distance. Such property will allow us to skip many expensive DTW calculations when the lower bound value is greater than the best-so-far distance. Regardless of many well proposed lower bounds [25, 26], we consider LB\_Keogh [14] in this work due to its simplicity and tightness of the bound. However, LB\_Keogh requires a DTW global constraint ( $r$ ) to construct an upper ( $U$ ) and lower ( $L$ ) envelopes around any sequence  $Q$ , as follows:

$$U_i = \max(q_{i-r} : q_{i+r}) \tag{6}$$

$$L_i = \min(q_{i-r} : q_{i+r}) \tag{7}$$

Once the envelope is constructed, the lower bound value between a sequence  $C$  and the envelope is computed using Eq. (8). Any part of the sequence beyond the envelope (above or below) will be included as part of the lower bound, as shown in Fig. 3.

$$LB\_Keogh(Q, C) = \sum_{i=1}^n \begin{cases} (U_i - c_i)^2 & \text{if } c_i > U_i \\ (L_i - c_i)^2 & \text{if } c_i < L_i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$



**Fig. 3.** An illustration of the upper and lower envelope in LB\_Keogh function. The lower bound value is computed by the sum of the squared distance between the upper or lower envelope and the sequence  $C$  that are outside the envelope area.

### 2.4 Support Vector Machine (SVM)

SVM is a powerful classification algorithm that has been applied to a variety of problems and domains, including finance and medicine [15, 16]. SVM is based on the idea of constructing a hyperplane to separate the data that may be mapped to higher dimensional feature space using a kernel function. In the other words, if the data can be linearly separated, a hyperplane is constructed from input data, which gives a maximum margin called support vector as shown in Fig. 4. Otherwise, a kernel function is applied to map

the input space into higher dimensional feature space that can be linearly separated by a hyperplane [17] as shown in Fig. 5.

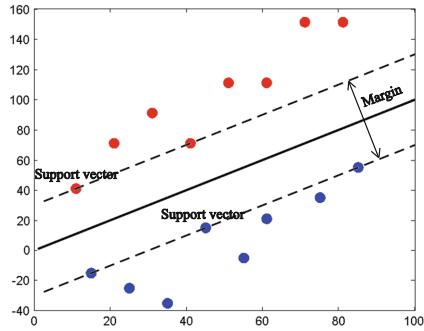


Fig. 4. Support vector machine with linearly separable data.

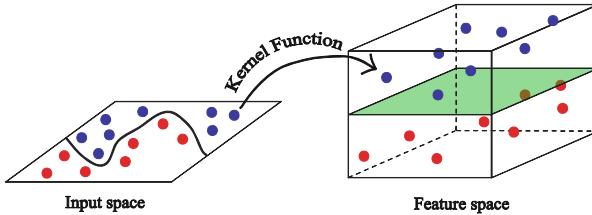


Fig. 5. Support vector machine uses a kernel function to map non-linearly separable data from input space into higher dimensional space that can be separated by a hyperplane.

### 2.5 Related Works

Some works have used DTW distance as an SVM's feature for time series classification problem [7–9]. Gudmundsson, S. et al. used DTW as a feature and concentrated on designing a complex SVM kernel so called ppfSVM [8], but the results turned out that it still could not beat 1-NN with DTW even though their SVM achieved better results over other existing SVM approaches. Most recently, Kate, R. J. found that employing a simple polynomial kernel function is sufficient with no need to design a complex kernel function [9]. Together with the use of other features such as Symbolic Aggregate approximation (SAX) [18], Euclidean distance, and DTW with global constraint, the results have showed to even outperform 1-NN using DTW with global constraint. However, regardless of its winning performance on accuracies, the time complexity is still its drawback. Therefore, this motivates us to aim at building a faster time series classifier while maintaining its robustness in achieving high classification accuracies.

### 3 Proposed Work and Experiment Setup

In this section, we provide details of our proposed work, which creates features based on LB\_Keogh values that will be used in SVM classifier. Then, we also provide information about datasets and tools used in the experiments.

#### 3.1 LB\_Keogh Feature Construction

As discussed in Sect. 2.5, instead of using DTW with global constraint distance as a feature, which is extremely expensive, LB\_Keogh with a global constraint is used to construct the feature. For each training data sequence, LB\_Keogh feature vector is constructed by computing lower bound values from that data sequence to every single sequence in the training set. Testing is done very similarly; LB\_Keogh feature vector is obtained from computing lower bound values from each of the test data to every training data sequence. More concretely, given a training dataset  $C$  with  $k$  time series sequences, a feature vector of any sequence  $Q$  that will be used in a classifier is defined as follows.

$$LB\_Keogh\_Feature(Q) = \{LB\_Keogh(Q, C_1), LB\_Keogh(Q, C_2), \dots, LB\_Keogh(Q, C_k)\} \quad (9)$$

#### 3.2 Making a Classifier

Once the LB\_Keogh feature is obtained, LIBSVM [20] was used to build an SVM classifier with the following parameters: SVM type = C-SVC, Penalty parameter  $C = 1$ , and Kernel type = polynomial with coefficient and gamma = 1. We also perform 10-fold cross validation on each training set to determine the polynomial degree that fits the best to the problem. After the classifier is built for each dataset, its performance is tested using its corresponding unseen test data.

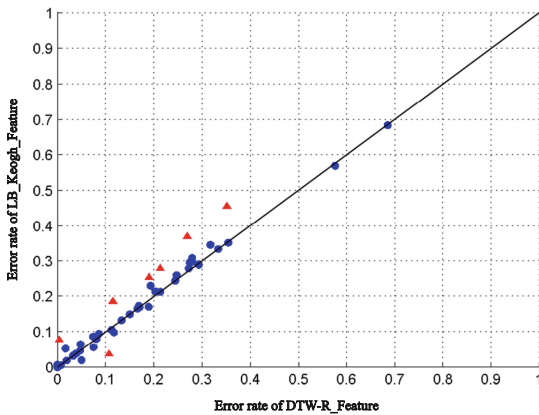
#### 3.3 Datasets and Experiment Setup

In this work, we employ the whole 47 benchmark datasets from the UCR Time Series Classification Archive [19]. The information about the datasets (e.g., the number of classes, sequence length, train/test split) can be found on the archive. For each dataset, the DTW's global constraint parameter  $r$  is set according to what is reported on the UCR Time Series Mining webpage [19]. The performance is evaluated by comparing classification error rates as well as the computational time in feature vector construction of our proposed work (LB\_Keogh\_Feature) with the state-of-the-art SVM classifier with DTW-R distance as the feature proposed by [9]. All of the experiments were carried out using Weka (v.3.6.12) software [21] on core i7-3770 3.40 GHz CPU with 16 GB of RAM.

## 4 Experiment Results and Discussion

Figure 6 visually compares the classification error rates between our proposed SVM using LB\_Keogh\_Feature and the most recently proposed SVM using DTW-R\_Feature.

Note that the points above the diagonal line denote the datasets on which the rival method DTW-R\_Feature performs better (with lower error rate), and the points below the diagonal line denote the datasets on which our proposed LB\_Keogh\_Feature performs better, achieving lower error rate. The red triangular points (7 out of 47 datasets) denote the discrepancies in the error rates that are statistically significant at 5 % level. We can see that the error rates between the two approaches are mostly very comparable, with the exception of the following six datasets, Cricket\_Y, FaceAll, FacesUCR, Lighting2, OSULeaf, and Two\_Patterns, where DTW-R\_Feature is better, and one dataset, CinC\_ECG\_torso, where DTW-R\_Feature is worse.



**Fig. 6.** A comparison between the classification error rates using our proposed LB\_Keogh\_Feature and the state-of-the-art DTW-R\_Feature on all 47 datasets. The red triangular points indicate the discrepancies in the error rates that are statistically significant at 5 % level (Color figure online).

LB\_Keogh\_Feature generally works very well on most datasets. However, we have looked into the feature vector of the six datasets where it performed statistically worse than DTW-R\_Feature. In particular, for each dataset, we sum up all the feature vectors in the training data and compare between the feature vectors from our proposed LB\_Keogh\_Feature and the rival DTW-R\_Feature. Figure 7 shows some snippets of the feature vectors in each class of three datasets, i.e., Car, Cricket\_Y, and Lighting2; the blue lines represent the LB\_Keogh\_Feature, and the red lines represent the DTW-R\_Feature. Note that both algorithms have the same error rate in Car dataset, as it shows very similar feature vectors, whereas DTW-R\_Feature outperforms in the other two datasets. In the latter case, we can see that since the features from LB\_Keogh are essentially lower bounds or approximation to the real DTW-R distance, the blue lines show that the LB\_Keogh\_Feature generally underestimates the distance, i.e., achieving not very tight lower bounds, which could in essence affect the classification accuracy.

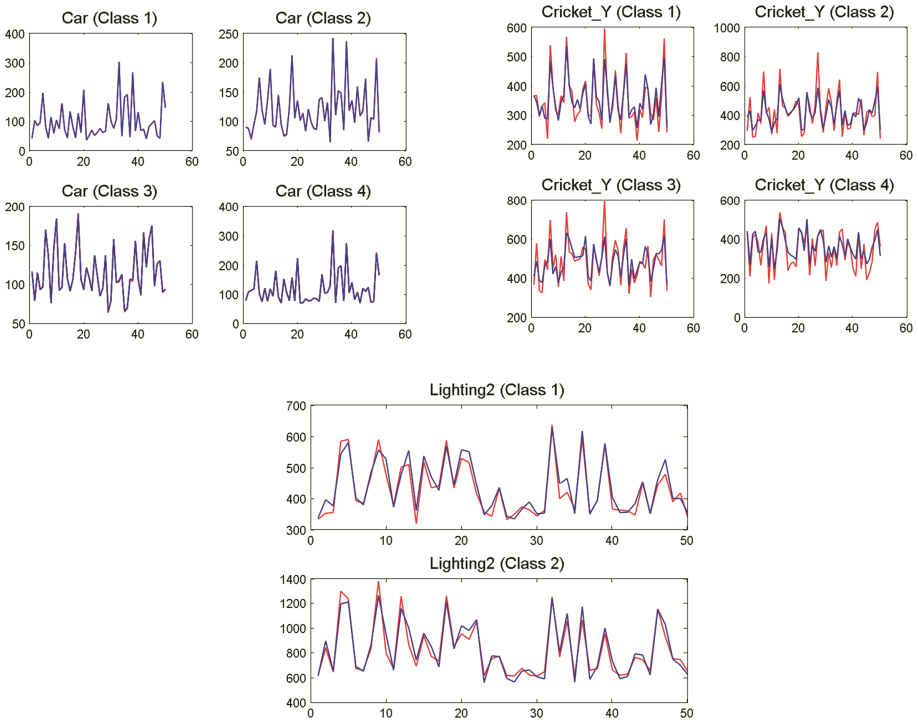


Fig. 7. Some snippets of the sum of the feature vectors within each class of the three training data, comparing between those from our proposed LB\_Keogh\_Feature in blue and those from the rival DTW-R\_Feature in red (Color figure online).

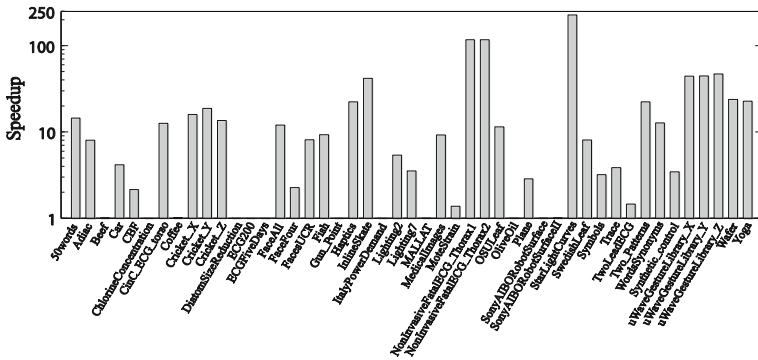


Fig. 8. The speedups of our proposed LB\_Keogh\_Feature over the rival method on all 47 datasets

The computational time in feature vector construction of both approaches are recorded. The speedups of our proposed LB\_Keogh\_Feature over the rival method are shown in Fig. 8. We can see that LB\_Keogh\_Feature vector is faster to be constructed in almost all datasets, except for 12 datasets whose global constraint values are zero



(i.e., Euclidean distance is performed instead of DTW), where both approaches consume similar time complexity. The average speedup of all 47 datasets is 19.79 (26.24 speedup if 12 Euclidean datasets are excluded).

## 5 Conclusion

In this work, we propose a fast time series classification based on SVM that used LB\_Keogh distance as a feature. The experiment results show that our proposed work can speed up the classification tasks by a large margin, while being able to maintain high accuracies, comparing with the state-of-the-art approach where the real DTW-R distance is used as a feature. It is also observed that our proposed feature can achieve comparable classification accuracies (little lower in some and little higher in some, with no statistical significance). However, in some datasets where the error rates drop down a little, statistically differing at 5 % significant level, good speedups on all the datasets are achieved, indicating the tradeoff between the classification accuracies and the running time it could save.

**Acknowledgements.** This research is partially supported by the Thailand Research Fund and Chulalongkorn University given through the Royal Golden Jubilee Ph.D. Program (PHD/0057/2557 to Thapanan Janyalikit) and CP Chulalongkorn Graduate Scholarship (to Phongsakorn Sathianwiriya).

## References

1. Sivaraks, H., Ratanamahatana, C.A.: Robust and accurate anomaly detection in ECG artifacts using time series motif discovery. *Comput. Math. Methods Med.* **2015**, 1–20 (2015)
2. Kurbalija, V., Radovanović, M., Ivanović, M., Schmidt, D., von Trzebiatowski, G.L., Burkhard, H.D., Hinrichs, C.: Time-series analysis in the medical domain: A study of Tacrolimus administration and influence on kidney graft function. *Comput. Biol. Med.* **50**, 19–31 (2014)
3. Zeng, Z., Yan, H.: Supervised classification of share price trends. *Inf. Sci.* **178**(20), 3943–3956 (2008)
4. Mahabal, A.A., Djorgovski, S.G., Drake, A.J., Donalek, C., Graham, M.J., Williams, R.D., Larson, S.: Discovery, classification, and scientific exploration of transient events from the Catalina Real-time Transient Survey. arXiv preprint (2011). [arXiv:1111.0313](https://arxiv.org/abs/1111.0313)
5. Bagnall, A., Lines, J.: An experimental evaluation of nearest neighbour time series classification. arXiv preprint (2014). [arXiv:1406.4757](https://arxiv.org/abs/1406.4757)
6. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
7. Jeong, Y.S., Jayaraman, R.: Support vector-based algorithms with weighted dynamic time warping kernel function for time series classification. *Knowl.-Based Syst.* **75**, 184–191 (2015)
8. Gudmundsson, S., Runarsson, T.P., Sigurdsson, S.: Support vector machines and dynamic time warping for time series. In: *IEEE International Joint Conference on Neural Networks, IJCNN 2008, (IEEE World Congress on Computational Intelligence)*, pp. 2772–2776. IEEE (2008)

9. Kate, R. J.: Using dynamic time warping distances as features for improved time series classification. *Data Min. Knowl. Discov.*, pp. 1–30 (2015)
10. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **26**(1), 43–49 (1978)
11. Niennattrakul, V., Ratanamahatana, C.A.: Learning DTW global constraint for time series classification. arXiv preprint (2009). [arXiv:0903.0041](https://arxiv.org/abs/0903.0041)
12. Park, S., Lee, D., Chu, W.W.: Fast retrieval of similar subsequences in long sequence databases. In: *Proceedings of the Workshop on Knowledge and Data Engineering Exchange, (KDEX 1999)*, pp. 60–67. IEEE (1999)
13. Kim, S.W., Park, S., Chu, W.W.: An index-based approach for similarity search supporting time warping in large sequence databases. In: *Proceedings of 17th International Conference on Data Engineering*, pp. 607–614. IEEE (2001)
14. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **7**(3), 358–386 (2005)
15. Zhiqiang, G., Huaiqing, W., Quan, L.: Financial time series forecasting using LPP and SVM optimized by PSO. *Soft. Comput.* **17**(5), 805–818 (2013)
16. Labate, D., Palamara, I., Mammone, N., Morabito, G., La Foresta, F., Morabito, F.C.: SVM classification of epileptic EEG recordings through multiscale permutation entropy. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5. IEEE (2013)
17. Awad, M., Khan, L., Bastani, F., Yen, I.: An effective support vector machines (SVMs) performance using hierarchical clustering. In: *16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004*, pp. 663–667. IEEE (2004)
18. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 2–11. ACM (2003)
19. The UCR Time Series Classification Archive. [http://www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/)
20. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2**(3), 27 (2011)
21. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *ACM SIGKDD Explor. Newsl.* **11**(1), 10–18 (2009)
22. Das, G., Gunopulos, D., Mannila, H.: Finding similar time series. In: Komorowski, J., Żytkow, J.M. (eds.) *PKDD 1997. LNCS*, vol. 1263, pp. 88–100. Springer, Heidelberg (1997)
23. Marteau, P.F.: Time warp edit distance with stiffness adjustment for time series matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(2), 306–318 (2009)
24. Chen, L., Ng, R.: On the marriage of lp-norms and edit distance. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, vol. 30, pp. 792–803. VLDB Endowment (2004)
25. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: FTW: fast similarity search under the time warping distance. In: *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 326–337. ACM (2005)
26. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Keogh, E.: Searching and mining trillions of time series subsequences under dynamic time warping. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 262–270. ACM (2012)