B. Mitschang et al. (Hrsg.): Datenbanksysteme für Business, Technologie und Web (BTW 2017),
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 289

Benchmarking Univariate Time Series Classifiers

Patrick Schäfer[1] Ulf Leser[2]

Abstract: Time series are a collection of values sequentially recorded over time. Nowadays, sensors for recording time series are omnipresent as RFID chips, wearables, smart homes, or event-based systems. Time series classification aims at predicting a class label for a time series whose label is unknown. Therefore, a classifier has to train a model using labeled samples. Classification time is a key challenge given new applications like event-based monitoring, real-time decision or streaming systems. This paper is the first benchmark that compares 12 state of the art time series classifiers based on prediction and classification times. We observed that most of the state-of-the-art classifiers require extensive train and classification times, and might not be applicable for these new applications.

Keywords: Benchmark, Time Series, Classification, Scalability.

1 Introduction

Time series (TS) are a collection of values sequentially recorded over time. TS are increasingly popular due to the growing importance of automatic sensors producing an every increasing flood of large, high-resolution TS in areas such as RFID chips, wearable sensors (wrist bands, smart phones), smart homes [JZ14], or event-based systems [MZJ13]. TS emerge in many applications, like weather observations, industry automation, mobility tracking, etc.

This study focusses on time series classification (TSC). TSC describes the task of predicting a class label for a TS whose label is unknown. Therefore, a classifier has to train a model using labeled TS. The UCR time series classification and clustering archive [Ch15] is a representative selection of TS datasets. Along with the release of these datasets, the authors published accuracy baselines to make TSC publications comparable. The largest UCR datasets contain a few thousand TS of a few thousand measurements. At the same time real-time decision systems emerge with billions of measurements for thousands of sensors. As a concrete example, seizures in long-term human intra-cranial EEG recordings of epilepsy patients have to be predicted [Pr15]. This dataset contains EEG recordings of 10 minutes each and accounts for more than 50GB with $240000 \times 16 \times 6000$ measurements from 6000 samples and 16 electrodes. As another real-world example, energy consumption profiles from smart plugs deployed in households [JZ14] were recorded. This dataset contains four billion measurements from 2125 plugs distributed across 40 households. It aims at load prediction and outlier detection using the power profiles of electrical devices. In such applications a key challenge is to provide scalability in runtime combined with a high classification accuracy.

[1] Humboldt Universität zu Berlin, Germany, patrick.schaefer@hu-berlin.de
[2] Humboldt Universität zu Berlin, Germany, leser@hu-berlin.de

An excellent survey of TSC algorithms is given in [Ba16]. Generally, one can observe a trade-off between accuracy and runtime of TSC algorithms. A trend in TSC is to build ensembles of core classifiers [Ba15, LB14]. While this does increase accuracy significantly, the runtime is negatively affected, as each core classifier has to be trained to build the ensemble and each has to predict a label. Another method is to reduce prediction times at the cost of training times or accuracy. For example, Dynamic Time Warping (DTW) typically considers windows of a certain length only, which both reduces runtime and improves accuracy, if the length limit is set properly [Pe14, LB14]. Similar ideas exist for other techniques such as shapelet classifiers. A runtime optimized version is Fast Shapelets (FS) [RK13].

A recent comparative study [Ba16] compares 18 recent TS classifiers in terms of classification accuracy. Classification accuracy has been the key metric to evaluate new TSC methods [Ba16, Di08, LB14]. This paper presents the first benchmark based on runtimes and accuracy for 12 state-of-the-art TS classifiers. We identified four groups of TS classifiers: whole series, shapelets, bag-of-features, and ensembles. For each group, we have selected the most accurate [Ba16] and fast representatives, if the implementation was available. Combined, our benchmark ran for more than 1000 CPU days. We observed that most of the state-of-the-art classifiers require extensive train and classification times.

The rest of the paper is structured as follows: Section 2 contains the background on TS classification and related work. Section 3 presents state of the art in TSC. An experimental evaluation is presented in Section 4.

2 Background & Related Work

2.1 Definitions

A univariate *time series* is a sequence of $n \epsilon \mathbb{N}$ real values, ordered in time. If the sampling rates of the TS are the same, one can omit the time stamps for simplicity sake and consider the TS as sequences of n-dimensional data points: $T = (t_1, \ldots, t_n), n \epsilon \mathbb{N}$. We associate each TS with a class label $y \epsilon \mathbb{N}$. All TS with the same label represent a class. *Time series classification (TSC)* describes the task of predicting a class label for a TS whose label is unknown.

2.2 UCR time series classification archive

The UCR time series classification and clustering archive [Ch15] is often used as basis for benchmarking and comparing in TS research. It contains a representative sample of univariate TS use cases. In its initial version, it contained 45 datasets. It has recently been expanded to 85 datasets. Each dataset is split into a train and test set. All time series of a dataset have the same length. These datasets include a vast variety of TS, including, motion sensors (inline-skating, gun aiming, cricket), ECG signals, spectrograms, starlight-curves,

and image outlines (anthropology, face or animal contours); 5 of these 85 datasets are synthetic. Some datasets have only a few dozen samples (Beet, Coffee, OliveOil), while the largest contain thousands of TS with thousands of measurements (StarLightCurves, FordA, FordB). In total, there are roughly 50.000 train TS and 100.000 test TS and a total of 55 million measurements.

2.3 Related Work

Previous comparative studies on TS classification had focussed mostly on the accuracy of TS classifiers. In [Di08] 8 TS representations and 9 whole series distance measures on a subset of 38 datasets from the UCR time series archive were benchmarked. They found that there was no whole series distance measure that was superior to others in terms of accuracy. In [LB14] elastic distance measures are compared on 75 TSC problems, with 46 datasets from the UCR archive. They found no significant difference between elastic distance measures and that through ensembling a more accurate classifier than each single core classifier can be created. A recent study [Ba16] compares 18 state-of-the-art classifiers in terms of accuracy, and runtimes have not been evaluated. They implemented all classifiers in a common JAVA framework on 85 UCR datasets. They found that 9 algorithms are significantly more accurate than the baselines (DTW and Rotation Forest). The COTE ensemble [Ba16] was the most accurate in their study. In [Sc16] we proposed a fast classifier and benchmarked its runtime against 5 competitors based on our own implementations or those given by the authors. With the implementation of the state-of-the-art classifiers in [Ba16], we are now in the position to extend our benchmark with 12 state-of-the-art classifiers in terms of accuracy *and runtime*.

3 Approaches to Time Series Classification

TS classifiers can be divided into four groups.

Whole Series: TS are compared by a distance measure applied to the whole TS data. Elastic distance measures compensate for small differences in the TS such as warping in the time axis. They are ill-suited if only TS subsequences are important as in all EEG or ECG signals.

Shapelets: Shapelets are subsequences of a TS that are maximally representative of a class label, and can appear at any offset. A TS can be assigned to a classes by the absence of, presence of, or the Euclidean distance to a shapelet.

Bag-of-Features / Bag-of-Patterns: These approaches distinguish TS by the frequency of occurrence of subsequences rather than their presence of absence. Firstly, subsequences are extracted from TS. Secondly, features are generated from these subsequences, i.e., by generating statistics over or discretization of the subsequences. Finally, these feature vectors are used as input to standard classifiers.

Ensembles: Ensembles combine different core classifiers (shapelets, bag-of-patterns, whole series) into a single classifier. Each core classifier produces a label and a (majority) vote is performed. These classifiers have shown to be highly accurate at the cost of an increased runtime [Ba16].

3.1 Whole Series

Dynamic Time Warping (DTW) [Ra12]: DTW is an elastic similarity measure as opposed to the Euclidean distance (ED). DTW calculates the optimal match between two TS, given some restraints on the amount of allowed displacement of the time axis. This best warping window size is typically learned by cross validation on a training set. This provides warping invariance and essentially is a peak-to-peak and valley-to-valley alignment of two TS. It is likely to fail if there is a variable number of peaks and valleys in two TS. DTW is commonly used as the baseline to compare to [Di08, LB14, Ba16]. Early abandoning techniques and cascading lower bounds have been introduced in [Ra12], which we implemented for our runtime benchmark.

3.2 Shapelet Approaches

Fast Shapelets (FS) [RK13]: Finding shapelets from a set of TS is very time consuming. Subsequences of variable length have to be extracted at each possible offset of a TS and the distance of the subsequences to all other TS is minimized. These subsequences whose distance best separates between classes are used as shapelets. To speed up shapelet discovery, the FS approach makes use of approximation and random projections. Each candidate is discretized and the word count is stored. Then multiple random projections are generated to allow for single character flips in a word. The frequency of a word after projection approximates the occurrences of a subsequence within all TS. The top k words that best separate between classes are mapped back to the TS subsequences. These subsequences represent the nodes of a decision tree. The distance to each subsequence (shapelet) is used as a branching criterion.

Shapelet Transform (ST) [BB15]: The ST separates the shapelet discovery from the classification step. First, the top k shapelets are extracted from the data. Next, the distance of each TS to all k shapelets is computed to form a new feature space. Finally, standard classifiers such as Naive Bayes, C4.5 decision trees, SVMs, Random Forests, Rotation Forests and Bayesian networks are trained with this feature space. Each classifier is assigned a weight based on the train performance with the aim to build an ensemble using a weighted vote for prediction. ST is the most accurate shapelet approach according to an extensive evaluation in [Ba16].

Learning Shapelets (LS) [Gr14]: In LS the shapelets are synthetically generated as part of an optimization problem, as opposed to extracting them from the samples as in ST or FS. The expressive power of this model is much better, as the algorithm can generate smoothed versions of the subsequences or subsequences that do not exist within the data. The shapelets

are initialized using k-means clustering. This method then uses gradient descent and a logistic regression model to jointly learn the weights of the model and the optimal shapelets.

3.3 Bag-of-Features / Bag-of-Patterns Approaches

Time Series Bag of Features (TSBF) [BRT13]: The TSBF approach extracts random subsequences of random lengths from the TS. It then partitions these into shorter intervals and statistical features are extracted. A codebook is generated with these features using a random forest classifier. Finally, a supervised learner is trained with the codebook.

BoP [LKL12]: The BoP model extracts sliding windows from a TS and discretizes these windows to words using Symbolic Aggregate approXimation (SAX) [Li07]. The best window size has to be learned from training. Discretization is performed by calculating mean values over disjoint subsections of a window. Each mean value is then discretized to a symbol using equi-probable intervals. The frequency of words is recorded in a histogram. The Euclidean distance (ED) between two histograms is used for similarity, which represents the difference in word frequencies.

SAX VSM [SM13]: SAX VSM is based on the BoP approach. However, it uses a tf-idf representation of the histograms. A histogram is built for each class, as opposed to each TS in BoP. Words that occur frequently across all classes obtain a low weight, whereas words unique within a single class obtain a high weight. The Cosine similarity between a TS histogram and the tf-idf class vectors is used for similarity. The use of a tf-idf model instead of 1-nearest neighbour (1-NN) search reduces the computational complexity.

BOSS [Sc15]: A recent bag-of-patterns model is Bag-of-SFA-Symbols (BOSS). Sliding windows are extracted and each window is transformed into a word. In contrast to BoP, it makes use of the truncated Fourier transform and discretizes the real and imaginary parts of the Fourier coefficients to symbols. This discretization scheme is called Symbolic Fourier Approximation (SFA) [SH12]. Both, SAX and SFA have a noise reducing effect, BOSS by the use of the first Fourier coefficients (low-pass filter) and SAX by averaging subsections. BOSS uses an asymmetric distance measure in combination with 1-NN search: only those words that occur in the 1-NN TS query are considered, whereas all words that occur exclusively in the TS sample are ignored. To improve performance, multiple window sizes are ensembled to a single classifier. BOSS is the most accurate bag-of-patterns approach according to [Ba16].

BOSS VS [Sc16]: BOSS VS is based on the BOSS approach and trades accuracy for runtime. It builds a tf-idf representation on top of SFA histograms using the Cosine similarity as distance measure. BOSS VS trains an ensemble using $\sqrt{n}$ window sizes at equi-distance.

3.4 Ensembles

Elastic Ensemble (EE PROP) [LB14]: EE PROP is a combination of 11 nearest neighbour whole series classifiers, including DTW CV, DTW, LCSS, ED. A voting scheme weights each classifier according to its train accuracy.

Collective of Transformation Ensembles (COTE) [Ba15]: COTE is based on 35 different core classifiers in time, autocorrelation, power spectrum and shapelet domain. It is composed of the EE (PROP) ensemble and ST classifier. It is the most accurate TSC according to [Ba16].

4 Experiments

Datasets: We evaluated all TS classifiers using the UCR benchmark datasets [Ch15]. Each dataset provides a train and test split.

Classifiers: We evaluated the state-of-the-art TSC: COTE, EE PROP, BOSS, BOSS VS, BoP, SAX VSM, LS, FS, ST, TSBF, 1-NN DTW and 1-NN DTW CV with a warping window constraint.

Implementation: Where possible, we used the implementation given by the authors [BO16, RK13] or the implementations given by [Ba16]. For 1-NN DTW and 1- NN DTW CV we make use of the state-of-the-art lower bounding techniques [Ra12]. Multi-threaded code is available for BOSS and BOSS VS, but we have restricted all codes to use a single core. We used the standard parameters of each classifier in the experiments.

Machine: All experiments ran on a server running openSUSE with a XeonE7-4830 2.20GHz and 512GB RAM, using JAVA JDK x64 1.8.

4.1 Classification Accuracy

Fig. 1 shows a critical difference diagram over the average ranks of the classifiers as introduced in [De06]. The classifiers with the lowest (best) accumulated ranks are to the right of the plot. Our results are similar to the ones previously published in [Ba16], with two exceptions: the BOSS VS classifier was not part of their experiments, and we have used the original train/test splits rather than resampling.

Whole Series: The 1-NN DTW and 1-NN DTW CV are among the worst (highest) ranked classifiers in our evaluation.

Shapelets: While FS is optimized for performance, ST is optimized for accuracy. As such ST shows the second lowest (best) rank and FS has the highest (worst) rank. ST first extracts shapelets and then trains an ensemble of classifiers on top of the shapelet representation. This might be one reason, why it is more accurate than LS, which is based on one standard classifier.

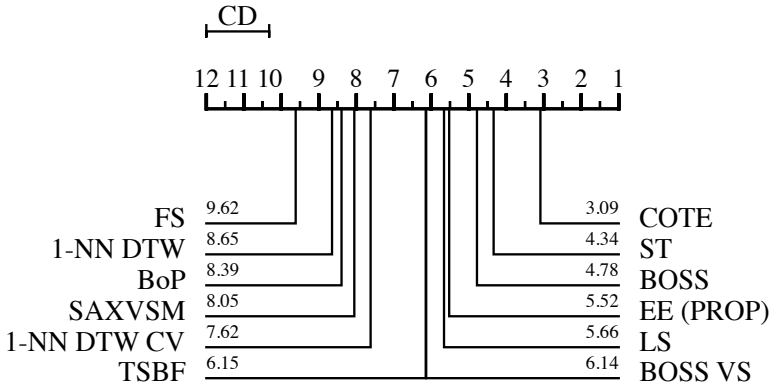


Fig. 1: Average ranks of state-of-the-art classifiers.

Bag-of-Patterns: BOSS and BOP are optimized for accuracy and BOSS VS and SAX VSM are optimized for speed. As such, BOSS shows the highest accuracy of these. Both TSBF and BOSS VS are more accurate than SAX VSM and BoP.

Ensembles: Ensemble classifiers have high accuracy at the cost of computational complexity. They offer a higher classification accuracy than each of the core classifiers, which are part of the ensembles. COTE is the most accurate classifier.

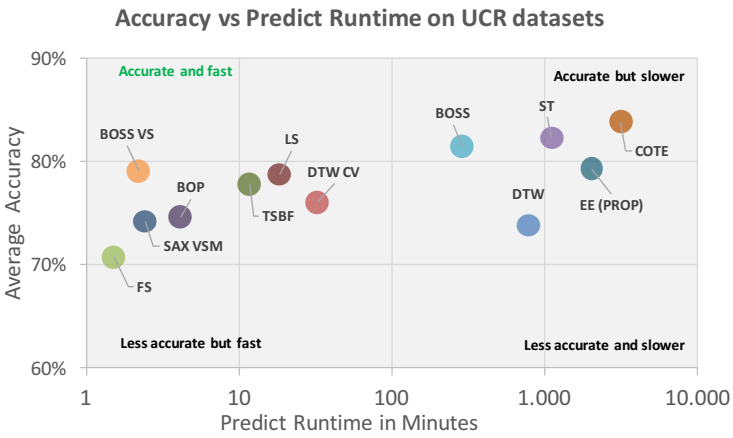
4.2 Runtime

Fig. 2 shows the CPU time on the x-axis (in logarithmic scale) and average accuracy on the y-axis for training (top) and prediction (bottom) of all 12 classifier on the datasets. Our experiments ran for more than 1000 CPU days, thus we had to limit the experiment to the 45 core UCR datasets, because of the high runtime of some classifiers, i.e., ST >1000 hours, EE (PROP) >1800 hours, and COTE >2900 hours for training using default parameters. EE and COTE results are still pending after 6 CPU weeks on the NonInvasiveFatalECGThorax1 and NonInvasiveFatalECGThorax2 datasets. All 45 UCR datasets account for roughly 17000 train and 62000 test TS in total.

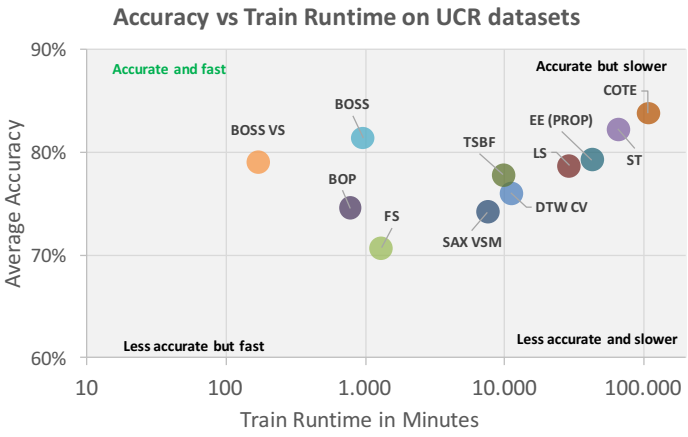
Whole Series: 1-NN DTW CV performs a training step that significantly reduces the runtime for the prediction step. Training DTW CV took 186 hours. DTW CV improves the average accuracy by some percent. Still, DTW CV and DTW show a rather low accuracy.

Shapelets: Shapelets are among the slowest (ST) and fastest classifiers (FS). If accuracy is important, ST is a good choice. If speed is important FS is a better choice. However, FS is the least accurate classifier in our experiments (compare Fig. 1). ST is more than 2 orders of magnitude slower than FS for prediction and training.

Bag-of-Patterns: BOSS VS is a runtime optimized version of BOSS, likewise SAX VSM is an optimized variant of BoP. BOSS VS shows a good trade-off between classification accuracy and runtime. It is orders of magnitude faster than most competitors, and equally



(a) Cumulative prediction (classification) time vs average accuracy. The runtime is in logarithmic scale.



(b) Cumulative train time vs average accuracy. The runtime is in logarithmic scale.

Fig. 2: Runtimes.

fast as FS while offering a much better accuracy. BoP and SAX VSM have rather high train times but fast test times. BOSS has the second highest accuracy and is faster than ST by one to two orders in magnitude.

Ensembles: EE PROP is an ensemble of whole series classifiers. As such it has a higher runtime but offers better accuracy than DTW and DTW CV, which are part of the ensemble. To obtain high accuracy, the COTE ensemble makes use of ST and EE PROP. Thus, its runtime is essentially a composition of these runtimes. Ensembles show the highest test and train runtime in the experiments.

In general, classifiers with high accuracy require time consuming training. By increasing train times, the prediction time can be reduced as for DTW and DTW CV. By ensembling classifiers, accuracy can be increased at the cost of runtime as for COTE, ST or EE (PROP). By sacrificing some accuracy a better runtime can be achieved as for BOSS VS, FS and SAX VSM. The authors of COTE, EE (PROP), and ST emphasize in [Ba16] that their code was not optimized for runtime and that each core classifier can be executed in parallel. However, we consider CPU time, which is independent of the number of cores used.

5 Conclusion

There is a trade off between classification accuracy and computational complexity. To obtain high accuracy, time series classifiers have to perform extensive training. For example the 1-NN DTW classifier can be used with and without a warping window constraint. When the constraint is set, the time for prediction is significantly reduced. However, the time to train the best window size prohibits its application in real-time and streaming contexts. By sacrificing some accuracy, the runtime of a classifier can be reduced by orders of magnitude. Overall, COTE, ST and BOSS show the highest classification accuracy at the cost of increased runtime. BOSS VS offers a good trade off between classification accuracy and runtime, as it is orders of magnitude faster than the most accurate classifiers. However, BOSS VS should be considered a starting point rather than the final solution. Future research in time series classification could lead to producing fast and accurate classifiers.

References

[Ba15] Bagnall, Anthony; Lines, Jason; Hills, Jon; Bostrom, Aaron: Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles. IEEE Transactions on Knowledge and Data Engineering, 27(9):2522–2535, 2015.

[Ba16] Bagnall, Anthony; Lines, Jason; Bostrom, Aaron; Large, James; Keogh, Eamonn: The Great Time Series Classification Bake Off: An Experimental Evaluation of Recently Proposed Algorithms. Extended Version. Data Mining and Knowledge Discovery, pp. 1–55, 2016.

[BB15] Bostrom, Aaron; Bagnall, Anthony: Binary shapelet transform for multiclass time series classification. In: International Conference on Big Data Analytics and Knowledge Discovery. Springer, pp. 257–269, 2015.

[BO16] BOSS implementation: . https://github.com/patrickzib/SFA/, 2016.

[BRT13] Baydogan, Mustafa Gokce; Runger, George; Tuv, Eugene: A bag-of-features framework to classify time series. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):2796–2802, 2013.

[Ch15] Chen, Y; Keogh, E; Hu, B; Begum, N; Bagnall, A; Mueen, A; Batista, G: , The UCR Time Series Classification Archive. http://www.cs.ucr.edu/~eamonn/time_series_data, 2015.

[De06] Demšar, Janez: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 7:1–30, 2006.

[Di08] Ding, Hui; Trajcevski, Goce; Scheuermann, Peter; Wang, Xiaoyue; Keogh, Eamonn: Querying and mining of time series data: experimental comparison of representationsx and distance measures. 2. VLDB Endowment, pp. 1542–1552, 2008.

[Gr14] Grabocka, Josif; Schilling, Nicolas; Wistuba, Martin; Schmidt-Thieme, Lars: Learning time-series shapelets. In: 2014 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 392–401, 2014.

[JZ14] Jerzak, Zbigniew; Ziekow, Holger: The DEBS 2014 Grand Challenge. In: 2014 ACM International Conference on Distributed Event-based Systems. ACM, pp. 266–269, 2014.

[LB14] Lines, Jason; Bagnall, Anthony: Time series classification with ensembles of elastic distance measures. Data Mining and Knowledge Discovery, 29(3):565–592, 2014.

[Li07] Lin, Jessica; Keogh, Eamonn J.; Wei, Li; Lonardi, Stefano: Experiencing SAX: a novel symbolic representation of time series. Data Mining and knowledge discovery, 15(2):107–144, 2007.

[LKL12] Lin, Jessica; Khade, Rohan; Li, Yuan: Rotation-invariant similarity in time series using bag-of-patterns representation. Journal of Intelligent Information Systems, 39(2):287–315, 2012.

[MZJ13] Mutschler, Christopher; Ziekow, Holger; Jerzak, Zbigniew: The DEBS 2013 grand challenge. In: 2013 ACM International Conference on Distributed Event-based Systems. ACM, pp. 289–294, 2013.

[Pe14] Petitjean, François; Forestier, Germain; Webb, Geoffrey I; Nicholson, Ann E; Chen, Yanping; Keogh, Eamonn: Dynamic Time Warping averaging of time series allows faster and more accurate classification. In: 2014 IEEE International Conference on Data Mining. IEEE, pp. 470–479, 2014.

[Pr15] Predict seizures in long-term human intracranial EEG recordings: . `https://www.kaggle.com/c/melbourne-university-seizure-prediction`, 2015.

[Ra12] Rakthanmanon, Thanawin; Campana, Bilson; Mueen, Abdullah; Batista, Gustavo; Westover, Brandon; Zhu, Qiang; Zakaria, Jesin; Keogh, Eamonn: Searching and mining trillions of time series subsequences under dynamic time warping. In: 2012 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 262–270, 2012.

[RK13] Rakthanmanon, Thanawin; Keogh, Eamonn: Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets. In: 2013 SIAM International Conference on Data Mining. SIAM, 2013.

[Sc15] Schäfer, Patrick: The BOSS is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery, 29(6):1505–1530, 2015.

[Sc16] Schäfer, Patrick: Scalable time series classification. Data Mining and Knowledge Discovery, 30(5):1273–1298, 2016.

[SH12] Schäfer, Patrick; Högqvist, Mikael: SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In: 2012 International Conference on Extending Database Technology. ACM, pp. 516–527, 2012.

[SM13] Senin, Pavel; Malinchik, Sergey: SAX-VSM: Interpretable time series classification using SAX and vector space model. In: 2013 IEEE International Conference on Data Mining. IEEE, pp. 1175–1180, 2013.