A Literature Survey of Early Time Series Classification and Deep Learning

Tiago Santos
Know-Center Graz
Inffeldgasse 13/6
Graz, Austria
tsantos@know-center.at

Roman Kern
Know-Center Graz
Inffeldgasse 13/6
Graz, Austria
rkern@know-center.at

ABSTRACT

This paper provides an overview of current literature on time series classification approaches, in particular of early time series classification.

A very common and effective time series classification approach is the 1-Nearest Neighbor classifier, with different distance measures such as the Euclidean or dynamic time warping distances. This paper starts by reviewing these baseline methods.

More recently, with the gain in popularity in the application of deep neural networks to the field of computer vision, research has focused on developing deep learning architectures for time series classification as well. The literature in the field of deep learning for time series classification has shown promising results.

Early time series classification aims to classify a time series with as few temporal observations as possible, while keeping the loss of classification accuracy at a minimum. Prominent early classification frameworks reviewed by this paper include, but are not limited to, ECTS, RelClass and ECDIRE. These works have shown that early time series classification may be feasible and performant, but they also show room for improvement.

CCS Concepts

•Mathematics of computing → Time series analysis; •Computing methodologies → Neural networks; Learning latent representations;

Keywords

time series classification, early time series classification, deep learning

1. INTRODUCTION

Time series arise wherever data is being collected and indexed by time. Its automated classification, i.e. the assignment of a certain label to a time series, is of immense importance. It is used in practice to solve problems such as using electrocardiography (ECG) data to assess whether a patient has some heart condition, identifying tendencies in financial stock markets, or even classifying sign language. Early time series classification, as the discipline focusing on arriving to a classification decision when observing only partial samples of a time series, is also critical for some application domains. Often cited examples of early classification of time series in practice include health diagnoses, where early classification of monitoring systems data may lead to the early detection and thus prevention of certain diseases, and predictive maintenance in an industrial setting, where the early identification of malfunctioning components in machines in a factory may prove invaluable for the manufacturing process.

This literature review focuses on past and current research on time series classification, with a focus on the early time series classification. It is structured as follows. A brief overview on definitions of time series and time series types to be considered by this paper serves as an introduction to the literature on this topic. Then, literature on general algorithms for time series classification is presented. The application of deep learning algorithms for time series classification is awarded a separate section. Finally, early time series classification literature is reviewed.

SamI40 workshop at i-KNOW '16 October 18–19, 2016, Graz, Austria

2. EARLY TIME SERIES CLASSIFICATION LITERATURE REVIEW

2.1 Time Series Definitions and Types

The author of [8] defines a time series as a series of observations x_t, with each observation x corresponding to a specific time t. They distinguish between *continuous* and *discrete* time series. The latter refers, according to the authors, to time series with a discrete set of observations, and the former to time series of continuously recorded observations over some time interval like $T_0 = [0, 1]$. The rate at which time series observations arrives, i.e. the sampling rate, can also vary. This paper focuses on literature regarding discrete time series with constant sampling rates.

Another distinguishing feature of time series regards the observations x_t. x_t may be defined as a singular value, in which case the time series is called a "univariate time series". If x_t represents an n-dimensional vector, then the time series is termed "multivariate time series". Both single variable and multivariate time series will be analyzed. The observation values are assumed to be defined in $\mathbb{R}$ and respectively in $\mathbb{R}^n$. Alternatively, these observation values may also stem from some set of symbols or alphabet Ω, a set with a finite number of elements. Such observed variables are termed *categorical*, but a review of literature on time series with categorical observation values will not be in the scope of this survey.

Time series can be seen in a large number of different contexts, such as economic forecasting, stock market analysis or inventory analysis, to name a few of the applications mentioned for example by [27]. Many other examples and applications of time series theory will be covered in more detail later.

Time series repositories contain samples of such time series for analysis purposes. The UCR Time Series Classification Archive by [11] includes not only a set of labeled univariate time series for classification, but also benchmarks for classifiers. Many papers addressing time series classification used this time series repository and its benchmarks as a data source and as a baseline, as will be seen later. One of the datasets cleaned and summarized in the collection by [11] is the Auslan dataset, which was originally prepared by [21]. This data set of multivariate time series consists of samples of Auslan (Australian Sign Language) signs, and it is also referenced by a number of time series classification papers.

2.2 Traditional Methods

2.2.1 *Definitions*

Time series classification refers to the process of assigning a label, or class, to a time series. There is a *very* large number of approaches to address the problem of time series classification. Thus, the following review of the literature on time series classification does not aim to be exhaustive, but to give an overview of common as well as cutting edge approaches and theory.

A distinction between time series classification and *early* time series classification is drawn here. Time series classification uses all available observations of a time series to assign a label to it. Early classification of time series, however, refers to time series classification using only a reduced amount of observations. The literature reviewed in this section concerns time series classification using the full time series length. More on early time series classification is presented in 2.4.

2.2.2 *1-Nearest Neighbor*

The definition given by [34] of the Nearest Neighbor Search problem in the field of computational geometry is applied here to the machine learning context: In a supervised classification setting, which assumes a train set of labeled samples, the 1-Nearest Neighbor algorithm takes a new, unlabeled instance and assigns to it the label of the nearest neighbor from that train set, according to a certain measure of distance in $\mathbb{R}$.

The definition of a distance measure is of central importance for time series comparison, and, thus, for the 1-Nearest Neighbor search too. This motivates literature on the usage of the 1-Nearest Neighbor search for time series classification to introduce new distance measures, as well as on optimize of well-established ones. The baseline distance measure is the Euclidean distance for univariate time series, and extensions thereof for multivariate time series, such as the Frobenius norm. Another widely used time series distance measure is Dynamic Time Warping, which aims to align out-of-phase time series with each other.

The paper [36] aims to improve the Euclidean distance 1-Nearest Neighbor's performance in a binary classification setting where few labeled data is available. The approach proposed by the authors involves reinforcing the 1-Nearest Neighbor classifier's performance on the unlabeled set by iteratively adding instances classified with high confidence to the train set, until some stopping criterion is achieved.

The 1-Nearest Neighbor classifier is used by [11], [41] and [36], among many others, to classify both univariate as well as multivariate time series. To name a few of the distance measures used with the 1-Nearest Neighbor algorithm, these papers respectively use the Euclidean distance and dynamic time warping for univariate time series, as well as the Frobenius distance for multivariate time series.

The authors of [11] use the 1-Nearest Neighbor algorithm with the Euclidean distance as a baseline for comparison with dynamic time warping with different warping windows, since the latter generally outperforms the Euclidean distance, as the authors conclude.

On the other hand, [41] optimize the k-Nearest Neighbors search (incl. 1-Nearest Neighbor) for multivariate time series with an extension of the Frobenius distance termed "Eros". "Eros" applies the Frobenius distance to the singular value decomposition of the covariance matrices of 2 different multivariate time series in matrix-form. This helps with getting both matrices to have equal dimensions to compute the Frobenius distance, as well to capture the importance of the covariances of columns of the multivariate time series matrices, i.e. of the variables within the time series.

2.2.3 *Other time series classification approaches*

Many other approaches have been proposed to improve the strong performance and benefits of the 1-Nearest Neighbor algorithm outlined above.

Take for example [20]. That paper describes an architecture that, given raw data as multivariate time series, extracts events, clusters them and combines them back again with globally computed features on the raw data to produce train data for learning classification rules. Its implementation, called "TClass", uses k-means clustering and the naive Bayes learner for the steps previously described. According to the author, this approach leads not only to improved classification accuracy, but also higher understandability of the features used for classification.

Other comprehensible features used in time series classification are so-called "literals": simple statistics, like averages, computed over intervals of the time series. Both [31] as well as [30] apply AdaBoost, as presented in [15], on those literals. AdaBoost is an algorithm which linearly combines many such simple classifiers to one better performing classifier. The paper [31] uses adaboosted literals to cope with variable length time series and perform early classification, as will be seen in more detail later in this literature review. In [30], the same authors aim to improve previous results by considering more complex literal combinations to generate new features. They then apply, on that set of new features, support vector machines, both linear as well as with the Gaussian kernel, to achieve performance gains over the results outlined by other papers using 1- and k-Nearest Neighbors, on the previously mentioned Auslan data set, among others.

Another approach regarding feature extraction concerns shapelets. Shapelets are sub-sequences of the time series that allow for classification basing on local, phase-independent similarity in shape, according to [18]. They aim to maximally represent the class of a time series. The authors of [18], [25] and [42] use shapelets to derive easy-to-interpret features, while also experimentally improving accuracy of the 1-Nearest Neighbor algorithm with the dynamic time warping distance (in some of the datasets of [11]).

Both the shapelet as well as the time series forest approaches are designed for univariate time series. One further approach that considers multivariate time series, by [4], combines the dynamic time warping distance with principal component analysis to create the "CBDTW" (correlation based dynamic time warping) measure. That similarity measure is computed by first segmenting an unclassified time series using principal component analysis, mapping the segments to the real numbers using a cost function and then applying the dynamic time warping distance to compare the unclassified time series to the train set of previously segmented time series. This approach thus leverages correlation effects to accurately describe time series classes.

All of the papers mentioned before use either the datasets of [11] or the multivariate time series data set of [21] or both. This makes benchmarks and comparisons between them feasible, effectively addressing concerns mentioned by [22].

The importance of hidden correlations in time series data, as highlighted by the paper [22], is one of the factors motivating the application of deep learning for time series classification.

2.3 Deep Learning for Time Series

2.3.1 Definitions

Recently, deep learning has also been applied to the time series classification problem. This section starts with a review of literature on deep learning in general and then addresses state-of-the-art deep learning approaches for time series classification.

To cite [32], quote, "A standard neural network (NN) consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations". There are input neurons, which get activated from the environment, and other neurons, for example hidden or output neurons, which, according to the same author, quote, "get activated through weighted connections from previously active neurons". The assignment of weights for the connections controls the output of the neural network. In this context, the process of tuning weights to attain certain output is termed learning. The neurons are typically grouped into layers called input, output or hidden (i.e. those between input and output) layers. Each layer transforms, often non-linearly, the aggregate activation of the previous layer and propagates that to further layers. Deep learning consists of assigning weights (in the context described above) across multiple such layers of often non-linear transformations.

As a part of artificial intelligence, deep learning techniques are currently experiencing both numerous practical applications as well as various research developments. The books [6] and [14] outline, in their reviews of deep learning methods and architectures, many of the types of deep learning models and their purposes. A few examples of deep neural network architectures used in supervised learning settings include multi-layer neural networks as explained above and convolutional neural networks.

The latter consists of a deep neural network, especially designed for computer vision tasks. As described by [24], convolutional neural networks have unique properties like "sparse connectivity", which means that each layer is associated with just one region of an input image, i.e. the so-called "receptive fields", and "shared weights", which refers to each layer having the same set of weights (but with different receptive fields). There are also several deep neural network architectures designed to work in an unsupervised context, such as Restricted Boltzmann Machines, a type of stochastic artificial neural network, Deep Belief Networks, which consist of multiple learning layers (like Restricted Boltzmann Machines) which are trained greedily per layer, or Autoencoders, a type of feed forward network designed to replicate its input.

Deep learning architectures like the ones described above have been applied to successfully address problems like speech recognition [19], image classification [35] or even beating professional Go players [33]. Besides the very promising practical results achieved in those areas, the author [6] mentions the ability deep architectures have to succinctly represent functions as opposed to very large shallow architectures, as the main theoretical advantage of deep learning. Furthermore, the broad availability of open-source software frameworks for deep learning eases the deployment of distributed, performant, complex and state-of-the-art deep neural network models. Notable examples thereof are Theano [7], Torch [12], Google's TensorFlow [1] and Microsoft's CNTK [2].

2.3.2 Deep learning for time series classification

Yet another prominent application of deep learning architectures lies in time series classification. The literature on this topic is reviewed next. One of the first references on multivariate time series classification with neural networks is [10]. That paper proposes a feedforward neural network for predicting flour prices for a number of geographical locations. That neural network's predictions outperform those by an autoregressive moving average model on the root mean squared error measure. However, this paper does not employ deep neural nets since it makes use of only one input layer, one hidden layer and one output layer.

The work by [3] also suggests that neural networks may bring performance improvements to time series forecasting. In their empirical study, the authors directly compare machine learning methods like support vector regression, k-Nearest Neighbor regression and multilayer perceptron to assess their performance as time series forecasting algorithms. The authors conclude that the multilayer perceptron is among the more accurate methods for that task. A multilayer perceptron is a simple neural network consisting of just one hidden layer (and, of course, input and output layers), so, like before, this paper also does not address the use of deep neural networks.

In [9] however, several deep neural networks are trained and analyzed for the task of energy demand load forecasting. A deep recurrent neural network of 2 hidden layers and delivered the best performance in terms of root mean squared error of predicted values. The authors also stress the importance of feature selection and engineering to fully tap neural networks' power to fit highly non-linear models.

To that end, both domain knowledge as well as, among other transformations, principal component analysis proved useful in achieving the best possible results.

Feature selection and engineering as part of the application of deep neural networks to time series data is also one of the topics of [23]. That work addresses the use of deep neural networks to derive, from raw data, relevant features for time series modeling in an unsupervised setting.

Finally, [5] proposes a complex model, consisting of a deep belief network coupled to a multilayer perceptron, to compose portfolios of stocks. In that work, the deep neural network's input are carefully selected stock value time indexes, which reflects the role domain knowledge plays in the architecture of complex deep neural networks. The deep neural network delivered promising results and performed better in comparison with, among others, a logistic regression network.

After dealing with the literature on time series classification, both with general as well as with deep learning algorithms, work on early time series classification will be reviewed next.

2.4 Early Time Series Classification

2.4.1 Definitions

The authors of [38], [28] and [13] all agree on the basic early classification definition: It is the problem of trying to come to a classification decision with as little observations of a time series as possible, while sacrificing classification accuracy as little as possible.

A different interpretation of the early classification problem is provided by [29], which tackles the computational performance side of time series classification. This paper improves both the time and resource complexity of 1-Nearest Neighbor with the dynamic time warping distance by creating nearest "centroid" classifiers that are both faster and at least as accurate as nearest neighbor algorithms.

Again, all of the papers mentioned below use either the datasets of [11] or the multivariate time series data set of [21] or both, which is once again helpful for benchmarking and comparing the approaches, as referred by [22].

2.4.2 Early works on the early time series classification problem

One of the first works on the topic of early classification, as defined over time series length, was written by [31]. The authors start with literals, which are, as mentioned before, simple indicators or statistics, for example if a time series is going up or down over a previously defined interval. These base literals are then combined with an AdaBoost.MH, a version of the prominent ensemble classifier created by [15], which can cope with multivariate time series with variable length. Since their approach can work with time series of variable length, if a partial time series is used as input to their ensemble classifier, some of the literals, though not all, will still be able to output a classification response. Using the fact that the ensemble classifier is just a linear combination of those literals, the authors omit literals with unknown values to still reach a classification decision on a partial sample of the time series, thus performing early classification. For the datasets CBF, Control, Trace and Auslan (also known in the paper as Gloves), the authors report a minimum classification error rate of 0.45%, on time series reduced to 80% of their original length, and a maximum of 83%, on time series reduced to 60% of its original length. The best results overall were achieved with boosted interval literals, which boast average error rates of 15% for the 60% time series length case and just 1.65% for the 80% case. While many of the datasets used by the paper are still available in [11], the Auslan / Gloves dataset (among others) has been updated since, thus rendering direct comparisons with current literature harder to make. Nonetheless, this paper's experimental evaluation of classification performance on an early version of the dataset by [11] reveals that early classification may be a promising future research avenue, for example by tuning the AdaBoost learning process to address early classification.

After [31], many other authors addressed and further formalized the early classification problem. With [37], [38] and [40], that group of authors advanced research on early classification of time series from a number of different perspectives.

The paper [37] presents a first approach dedicated to early classification basing on both sequential rules mining and sequential decision trees. The focus of this paper is, however, sequences, which are time series taking values from a finite set (like an alphabet). A more general context, which would be time series taking values in $\mathbb{R}$, is considered in the following papers.

2.4.3 ECTS

Further, more prominent early classification approaches include "Early Time Series Classification", or ECTS for short, developed firstly in [38] and described in further detail in [39]. The author stipulates desirable, additional characteristics for early classifiers in general. These are the "seriality" of early classifiers, i.e. the invariance of the classification decision once a certain length of the time series has been observed, and that its accuracy when classifying the reduced time series is retained (or at least does not drop too much) with respect to classification on the time series with full length. These properties guide the authors' derivation of the Early Time Series Classification framework. That framework builds upon the 1-Nearest Neighbor algorithm and introduces concepts like minimum prediction length and a clustering algorithm. Those concepts together enable smart grouping of time series in a train set according to common prefixes between those time series for early, serial and reliable classification. That framework outperforms not only weaker early classifiers such as 1-Nearest Neighbor fixed, which refers to the application of the 1-Nearest Neighbor on a time series with a fixed reduced length, but also even the 1-Nearest Neighbor algorithm with the full length time series, on some cases. The classification accuracy of the paper's algorithm is at least 85% for as little as about 47% of the original time series length, thus keeping the framework's performance always on par with that of the 1-Nearest Neighbor, despite the reduced time series lengths.

The problem of finding appropriate features for early time series classification is the main topic of [40], the last paper by those authors on early classification.

2.4.4 RelClass

More recently, [28] proposed a probabilistic framework, termed "RelClass", using quadratic discriminants and support vector machines for performing early classification. That paper's key concept is the reliability of the classification decision, i.e. the degree of confidence with which one can say that the current incomplete data is sufficient to come to the same classification as the complete data (with high probability). That framework allows for classification as soon as a reliability threshold is met. The results achieved by that framework compare favorably against the previously mentioned 1-Nearest Neighbor fixed and Early Classification of Time Series methods, both on classification accuracy and earliness. In general, results strongly depend on the dataset, varying from a low in classification accuracy of 27% on 87% of original time series length to a classification accuracy high of 99% for just 30% of time series length.

2.4.5 ECDIRE

The latest development in the theoretical treatment of early time series classification seems to be the paper by [26]. The authors employ a method called "Early classification framework for time series based on class discriminativeness and reliability of predictions", in short "ECDIRE", to classify bird songs early and also beat the results achieved by [38] and [28] on the same data sets used by those papers. "ECDIRE" is a probabilistic classification framework that is organized into 4 steps. The first step, termed by the authors "Analysis of the discriminativeness of the classes", aims to derive a set of time series timestamps which enable good discrimination of time series classes early. This step begins with the definition of a set of early time series timestamps as a percentage of total time series length. Then, a variation of a Gaussian Process classifier, which in turn is a Bayesian probabilistic classifier, is trained with cross-folding on each of the time series reduced by the previously defined percentages. Early timestamps are thus identified as those which still maintain an accuracy above the reduced time series length (in percentage) times the accuracy (also in percentage) attained with the same classifier on the full length time series. In a second step, thresholds for classification reliability are defined as a, quote, "distance in terms of differences in class probabilities of the winning class and the next most probable class" per time series and class. This improves reliability by avoiding uncertain predictions and thus also classification timestamps, which might be too early to distinguish certain classes from each other. Finally, the actual probabilistic classifiers are trained on the full training set with the early timestamps computed in step 1 (as well as with the full time series length, as a fallback solution if no early timestamp was found in some particular case). This procedure resulted in a framework which Pareto dominated the results obtained in the papers by [39] and [28] much more often than the other way round.

The paper [26] was not the only one to apply early classification techniques to address a real-world data sets and problems. Examples thereof include, but are not limited to, the following papers. The authors of [16] compose, in their paper, a hidden Markov model with a support vector machine to create an early classifier, which performed well on a medical domain dataset containing gene expression values for a number of multiple sclerosis patients, for which not much data was available. They report classification accuracies as high as 87% for as little as 43% of the original time series length. The work by [17] introduces a classifier with a reject option, which allows the classifier to abstain from a classification decision if it is not clear-cut, weighing in that with the cost of making further observations. Their classifier with the reject option is general enough to allow for different classification algorithms, like a support vector machine or k-Nearest Neighbors, to be used. This early classifier was used in the classification of odors, serving as the basis for a performant electronic nose.

3. CONCLUSIONS

In this paper, literature on time series definitions, its classification, in particular with deep neural networks, and early classification techniques were reviewed. In conclusion, there is a very large number of both theoretical as well as practical work to classify time series, and, in particular, classify time series early. In particular, this body of work presents not only theoretical results and frameworks to use for different types of time series and data sets, but it also provides results to benchmark against. The results of the early time series classification literature highlighted above, in particular of the paper by [26], have shown that early classification works on both theoretical as well as applied problems, with early time series classifiers showing classification accuracies on reduced time series on par with those obtained via the 1-Nearest Neighbor classifier equipped with the Euclidean distance.

4. REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] A. Agarwal, E. Akchurin, C. Basoglu, G. Chen, S. Cyphers, J. Droppo, A. Eversole, B. Guenter, M. Hillebrand, T. R. Hoens, X. Huang, Z. Huang, V. Ivanov, A. Kamenev, P. Kranen, O. Kuchaiev, W. Manousek, A. May, B. Mitra, O. Nano, G. Navarro, A. Orlov, H. Parthasarathi, B. Peng, M. Radmilac, A. Reznichenko, F. Seide, M. L. Seltzer, M. Slaney, A. Stolcke, H. Wang, Y. Wang, K. Yao, D. Yu, Y. Zhang, and G. Zweig. An introduction to computational networks and the computational network toolkit, 2014.

[3] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[4] Z. Banko and J. Abonyi. Correlation based dynamic time warping of multivariate time series. *Expert Systems with Applications*, 39(17):12814–12823, 2012.

[5] B. Batres-Estrada. Deep learning for multivariate financial time series. 2015.

[6] Y. Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[7] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[8] P. Brockwell and R. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer New York, 2013.

[9] E. Busseti, I. Osband, and S. Wong. Deep learning for time series modeling. Technical report, Technical report, Stanford University, 2012.

[10] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka. Forecasting the behavior of multivariate time series using neural networks. *Neural networks*, 5(6):961–970, 1992.

[11] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The ucr time series classification archive, July 2015.

[12] R. Collobert, C. Farabet, K. Kavukcuoglu, and S. Chintala. torch - a scientific computing framework for luajit, May 2016.

[13] A. Dachraoui, A. Bondu, and A. Cornuejols. Early classification of time series as a non myopic sequential decision making problem. In *Machine Learning and Knowledge Discovery in Databases*, pages 433–447. Springer, 2015.

[14] L. Deng and D. Yu. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, 2014.

[15] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[16] M. F. Ghalwash, D. Ramljak, and Z. Obradovic. Early classification of multivariate time series using a hybrid hmm/svm model. In *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

[17] N. Hatami and C. Chira. Classifiers with a reject option for early time-series classification. In *Computational Intelligence and Ensemble Learning (CIEL), 2013 IEEE Symposium on*, pages 9–16. IEEE, 2013.

[18] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881, 2014.

[19] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[20] M. W. Kadous. Learning comprehensible descriptions of multivariate time series. In *ICML*, pages 454–463, 1999.

[21] M. W. Kadous. Temporal classification: Extending the classification paradigm to multivariate time series. School of Computer Science and Engineering, University of New South Wales, 2002.

[22] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery*, 7(4):349–371, 2003.

[23] M. Langkvist, L. Karlsson, and A. Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] J. Lines, L. M. Davis, J. Hills, and A. Bagnall. A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–297. ACM, 2012.

[26] U. Mori, A. Mendiburu, E. Keogh, and J. A. Lozano. Reliable early classification of time series based on discriminating the classes over time. *Data Mining and Knowledge Discovery*, pages 1–31, 2016.

[27] NIST/SEMATECH. e-handbook of statistical methods - introduction to time series analysis, Oct. 2013.

[28] N. Parrish, H. S. Anderson, M. R. Gupta, and D. Y. Hsiao. Classifying with confidence from incomplete information. *The Journal of Machine Learning Research*, 14(1):3561–3589, 2013.

[29] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 470–479. IEEE, 2014.

[30] J. J. Rodriguez, C. J. Alonso, and J. A. Maestro. Support vector machines of interval-based features for time series classification. *Knowledge-Based Systems*, 18(4):171–178, 2005.

[31] J. J. Rodriguez, J. J. R. Guez, and C. J. Alonso. Boosting interval-based literals: Variable length and early classification. 2002.

[32] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[33] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[34] S. S. Skiena. *The algorithm design manual: Text*, volume 1. Springer Science & Business Media, 1998.

[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[36] L. Wei and E. Keogh. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753. ACM, 2006.

[37] Z. Xing, J. Pei, G. Dong, and S. Y. Philip. Mining sequence classifiers for early prediction. In *SDM*, pages 644–655. SIAM, 2008.

[38] Z. Xing, J. Pei, and S. Y. Philip. Early prediction on

time series: A nearest neighbor approach. In *IJCAI*, pages 1297–1302. Citeseer, 2009.

[39] Z. Xing, J. Pei, and S. Y. Philip. Early classification on time series. *Knowledge and information systems*, 31(1):105–127, 2012.

[40] Z. Xing, J. Pei, S. Y. Philip, and K. Wang. Extracting interpretable features for early classification on time series. In *SDM*, volume 11, pages 247–258. SIAM, 2011.

[41] K. Yang and C. Shahabi. An efficient k nearest neighbor search for multivariate time series. *Information and Computation*, 205(1):65–98, 2007.

[42] L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM, 2009.