Proposal for the Bachelor Thesis: Open Time Series Classification with TEASER

Jonas Pfeiffer

August 2022

Table of Contents

Introduction

Time series are sequences of timed real values, also called data points as defined in [1]. These can occur in several variants in many different areas of applications. For instance, monitoring EEGs in health care [2], observation of the stock market on which time series also occur [3], monitoring smart home time series data [4] and so on [5]. The time series collected this way are then to be automatically classified (e.g., the time series data collected can indicate that a machine needs maintenance or not) and often these decisions are time critical. Thus, in many applications not only the classification is relevant but also the earliest possible time stamp at which a correct classification can be

made. The *early time series classification* (ECTS) problem tackles the problem to make a prediction after looking at as few measurements as possible and achieving the highest possible accuracy at the same time. For solving this problem several Algorithms have been published. Some state-of-the-art examples are ECONOMY-K [6] or TEASER [1]. The latter is one of the best algorithms regarding the tradeoff between earliness and accuracy [7] and will be considered more in detail in this thesis. It tackles the problem by following a two-tier approach which includes pairs of *slave classifiers* and *master classifiers* for increasing prefixes of the time series, also referred to as *snapshots*. The slave is a full-fledged classifier predicting a class probability for one snapshot of a specific length. Typically, the snapshot length grows evenly i.e., proportional to the time series length and each one has an associated pair of slave and master classifiers. TEASER uses WEASEL [8] as default slave classifier. The predicted probabilities are then passed to the master classifier whereupon this decides, if the probabilities are high enough for a final classification or the classification must be postponed to a later snapshot. As default master classifier, TEASER uses a one-class Support Vector Machine (ocSVM) [9]. However, TEASER, like other ECTS algorithms, is ill-suited for time series of indefinite length as written in the conclusion of [10]. This implies that if at test time a time series occurs that is longer than any previously seen at train time, TEASER is not able to use a slave-master-pair for the current snapshot length and thus cannot make a reliable prediction for this or any longer length. This is a problem if time series are shorter at train time than at test time such as in an online scenario and predictions are then limited by the length of the longest snapshot from training. Another problem is different lengths of the times series at train time. With different length, TEASER cannot set a global size for the snapshots and how they should grow over time. To sum up, the classic ECTS approaches like TEASER consider time series of equal length at test and train time. A solution would be a streaming approach in which a sliding window of fixed length learned at train time slides over the time series and predicts always using an equal length as input. In this case, historic data is no longer considered at test time. At train time, this allows for time series to be of finite but possible variable length. At test time, the learned sliding window can run infinitely in theory. In addition, TEASER with a sliding window offers the possibility of

assigning several classes to a time series at different points in time or of revoking an earlier decision with a later one [11]. In this thesis, the sliding-window approach will be implemented and examined in more detail with the help of TEASER. The result can perhaps be generalized by other algorithms. To compare the measurements, results from measurements from [1] [7] [11] should also be compared and data sets from other papers are used, such as the UCR data sets [12] used in [1] [11], to have a comparison. In the following I will go into more detail about TEASER, formulate what I will investigate and how this can be achieved.

Basics

Early Classification

As described above a time series is a sequence of real values, typically ordered by timestamp. Time series classification is a supervised learning task in which, given a training set of time series with the corresponding class labels, a model is learned that can predict class labels of new unlabeled time series. As an extension *early classification of time series* (ECTS) is learned with the help of such a training data set, but in addition, the prediction not only optimizes accuracy, but also tries to make predictions as early as possible, i.e., with as little data as possible. This involves looking at a prefix of the time series that is constantly increasing in size and trying to make a prediction on an as small of a section as possible, as Figure 1 shows. These increasing prefixes are also called snapshots i.e., a snapshot is a prefix of a time series available for classification. The snapshot considered by the classifier at test time can only grow to the maximum length of the time series seen at train time.

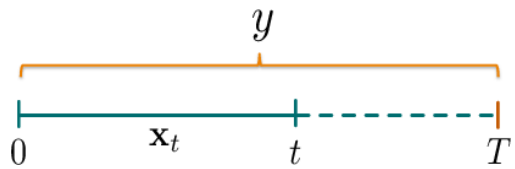

Figure 1: In ECTS the classifier sees the incoming snapshot x_t and predicts a label at every timestamp t [13].

In contrast, time series of indefinite length at test time are considered for *early classification of open time series* (ECOTS). This problem was considered in paper [13] for the first time. The paper defines the ECOTS problem, how it can be adapted from

the ECTS problem and how ECTS algorithms could be modified. Three main problems resp. differences between ECOTS and ECTS are identified. First, the ECOTS classifiers observe a sliding window instead of increasing snapshots, i.e., values of timestamps already seen may be discarded. The second problem is to adapt the earliness-accuracy-trade-off to the ECOTS problem and the third point, and the most interesting for this thesis, is the problem of finding a suitable size and step size for the sliding window[1].

TEASER

The Two-tier Early and Accurate Series classifier (TEASER) is a myopic approach for classifying time series as soon as possible. This means at each time step the classifier determines class probabilities and decides the time at which it seems right to make a prediction for the correct class. On the other hand, in a non-myopic approach the classifier is learning a decision time in the future which promises the best accuracy. The latter approach has the disadvantage that a time 0 is needed at which the classifier starts, like switching on a machine. Examples of algorithms that follow this approach are ECTS [14] and EDSC [15]. TEASER on the other hand can be evaluated in a time series starting at any time. In order to achieve this, it uses a two-tier approach consisting of pairs of slave and a master classifiers. The slave classifier computes class probabilities for a given snapshot. A snapshot is a fixed-length prefix of a time series. There are several snapshots considered per times series, each snapshot being a few timestamps longer than its predecessor. For each of these snapshots, a corresponding slave classifier calculates the class probabilities, this means that every slave classifier is associated with exactly one snapshot of fixed length. These probabilities are then passed on to the paired master classifier, which is trained on the confidence of prediction of its paired slave. If the masters have trusted the predictions of its associated slaves several snapshots in a row, a prediction is made. By default, WEASEL is used as the slave classifier and an ocSVM as the master in case of TEASER. The total number of snapshots and their length is determined by a user-defined interval length (e.g., 5, 10, 20) and the total length of the time

[1] In [1] a time series with a sliding window is defined as a data stream not so in [13], where a data stream is distinguished from the ECOTS by its concept shifts.

series. The interval length is needed because not every single incoming data point should trigger the slave classifier. Varying the number of snapshots leads to varying the number of master/slave-pairs which have an impact on the earliness-accuracy-trade-off and runtime. For early but possibly less accurate prediction a high number of snapshots is preferred and on the other hand if a high accuracy with inferior earliness is needed it is better to choose less snapshots. As already mentioned, the snapshot number and length depends on the maximum length of the time series in the training data. If there are shorter time series at train time, not all pairs of slave and master classifiers may be applied and on the other hand, if train time series are larger than the largest snapshot length, no suitable prediction can be made, apart from the naïve attempt to always use the last master/slave.

Objective

The aim of this thesis is to modify TEASER so that it uses sliding windows instead of snapshots. At train time, we need to find an optimal length for the sliding window and the step size. Thus, after train time, there will be only one pair of slave and master classifier, instead of multiple pairs for different snapshots lengths. This will solve both the problem that TEASER cannot deal with data sets of different length at train or test time. However, using a sliding window, thus discarding historic data could lead to inferior accuracy. For the mentioned step size, different length will be tested as a hyperparameter. The source code of TEASER must therefore be adapted so that it is able to solve the problem in the way described above. In the following potential data sets for testing and training are briefly presented, a few parameters are considered to enable a comparison and the ideas for adapting training and testing of TEASER to tackle the problem described above.

Data sets

In order to be able to compare the results of the measurements, it makes sense to use data sets that have already been used for similar problems. On the one hand, these would be data sets with which TEASER works in its basic version and data sets that were used in the investigation of problems that go in a similar direction as this one. For the latter, I will focus on the data sets

from [11]. In this thesis, datasets from the URA & UCR Time Series Classification Repository [12] were used, some of which were also used in TEASER. It makes sense to use these data sets again each contain a fixed train and test split. In addition, individual time series can still be manipulated in the train split, for example by hanging several time series one after the other to get time series of different length. In the test split, all data sets could even be directly attached to each other, as the streaming approach is designed for very long or arbitrarily long time series.

Experimental subjects

As it is still an ECTS problem despite the changed basic assumptions made in the training, the typical measurements are to be made. In concrete term this means the accuracy, earliness and harmonic mean of these two. This is particularly important in order to be able to draw any comparison at all with the other papers. The memory used and the runtime required during the training and test should also be recorded in order to detect changes between the original TEASER version and the adapted one.

Methods to be tested

As mentioned above, TEASER will be adapted to work with a sliding window instead of the growing snapshots, thus solving the problem of the different lengths of the time series at train time with the drawback of ignoring historic data, as well as offering the possibility to look at time series of any length at test time. At train time, TEASER should therefore no longer learn pairs of slave and master classifiers, which always observe a snapshot of a certain length, but instead learn a window length for which there is then only one pair of a slave and a master classifier. This means that there is no longer any need to set how the time series are to be divided according to their lengths i.e., by how much the snapshots are grown per classification. The problem also arises in a similar form with the sliding window, since the question arises in which time steps a classification should take place. Various settings should be considered for the step size here: On the one hand, the time window can be sliding forward by exactly its own length, so that disjunctive subsections of the time series are always considered for classification, or on the other hand the moving time window overlaps with its predecessor. For the latter, different relative sizes for sliding the

window could be tested. For instance, new data points could always be added halfway through the length of the time window before a new classification is carried out. The smaller the time step that the window advances before a new classification, the more interesting it could be to expect several identical classifications from TEASER before a result is accepted. Counting of several successive votes for the same class as has been done in TEASER with snapshots, would be the naïve approach to solve the historic data problem at this point. But it could also be interesting if there are other ways to solve this problem more precisely. Since TEASER should not terminate after a classification has been made, but rather continue the sliding window, TEASER can now also classify time series to multiple classes at different points in time or discard previous classifications and consider the latest classification to be the correct one for hole time series.

References

[1] Schäfer and Leser, "TEASER: Early and Accurate Time Series Classification," *arxiv,* 16 08 2019.

[2] Chen, Lu, Shang and Xie, "Automated change-point detection of eeg signals based on structual time-series analysis," *IEEE Access,* 06 11 2019.

[3] Idrees, Alam and Agarwal, "A prediction approach for stock market volatility on time series data," *IEEE Access,* 17 01 2019.

[4] Aminikhanghahi, Wang, Cook, "Real-Time Change Point Detection with application to Smart Home Time Series Data," *IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 5,* pp. 1010-1023, 01 05 2019.

[5] Gupta, Gupta, Biswas and Dutta, "Approaches and Application of Early Classification of Time Series: A Review," *arxiv,* 15 10 2020.

[6] Dachraoui, Bondu and Cornuéjols, "Early classification of time series as a non myopic sequential decision making

problem," *European Conf. on Machine Learning and Knowledge Discovery in Databases,* pp. 606-660, 2017.

[7] Akasiadis, Kladis, Michelioudakis, Alevizos and Artikis, "Early Time Series Classification Algorithms: An Empirical Comparison," 03 03 2022.

[8] Schäfer and Leser, "Fast and Accurate Time Series Classification with WEASEL," *arXiv,* 27 01 2017.

[9] Schölkopf, Platt, Shawe-Taylor, Smola and Williamson, "Estimating th Support of a High-Dimensional Distribution," *Neural Computation,* 27 11 2000.

[10] B. G. K. L. K. Löning, "sktime: A unified interface for machines learning with time series," *arxiv,* 17 09 2019.

[11] Achenchabe, Bondu, Cornuéjois and Lemair, "Early and Revocable Time Series Classification," *arXiv,* 22 09 2021.

[12] "UEA & UCR time series classification repository," [Online]. Available: https://www.timeseriesclassification.com/.

[13] Achenchabe, Bondu, Cornuéjols and Lemaire, "ECOTS: Early Classification in Open Time Series," *arxiv,* 01 04 2022.

[14] Xing, Pei and Yu, "Early Classification on Time Series," *Knowledge and Information Systems vol.31 no.1,* pp. 105-127, 2012.

[15] Xing, Pei, Yu and Wang, "Extracting Interpretable Features for Early Classification on Time Series," *International Conference of Data Mining,* pp. 247-258, 2011.