Realtime Early Time Series Classification with TEASER

Bachelor Thesis Exposé

Paul Kalz

First supervisor: Prof. Dr. Ulf Leser
Second supervisor: tba.

Department of Computer Science
Humboldt-Universität zu Berlin
Germany
July 2024

Introduction

A time series (TS) is a sequence of real values, reffered to as data points, which were recorded at regular time intervals [1]. They are widely used in sectors as diverse as healthcare [2, 3], industrial process monitoring [4] and financial markets [5]. The problem of assigning a predefined class to a time series in order to identify noteworthy events or patterns is referred to as time series classification (TSC). An example of how TSC is used involves determining whether a signal from an seismic sensor represents a volcano-tectonic earthquake or a man-made explosion like quarry blasts or illegal bomb fishing [6].
In many real-world applications, the speed at which a correct classification can be achieved is as important as the accuracy of the classification itself. This demand has introduced a new challenge in accurately classifying time series at the earliest possible point in time, known as early time series classification (ETSC). This problem has been studied extensively and several approaches and techniques have been developed, with many applications in various fields [7]. For example, in the medical field, to quickly identify the medical condition of a comatose patient using EEG data [8].
The concept of earliness in ETSC refers to predicting the correct class of a TS after considering only a minimum number of data points. To find a good solution to the ETSC problem, both earliness and accuracy need to be maximised. However, these objectives are inherently contradictory, as early classification relies on limited data which usually makes an accurate prediction more difficult. Conversely, having access to a larger data set allows for more accurate predictions, which is achieved by delaying the classification resulting in reduced earliness.
In ETSC, prior work has mostly considered earliness in terms of the number of data points used, but in the real world a fast prediction is essential. In real-time (wall-clock time) the number of data points per unit of time depends on the transmission frequency at which the data is received. An ETSC algorithm designed to work in real-time must consider the runtime of a classification and the data rate of the incoming data in order to work in a time-efficient manner, avoid building up a data backlog and deliver a classification result quickly. This new problem of finding an ETSC solution in real-time, which is the subject of this work, will be called Real-time Early Time Series Classification (RETSC).
In other works the RETSC Problem was also identified. The TEASER paper mentions an extension of their framework for fast classification in terms of wall-clock-time as a next step [1]. In a recent publication of an evaluation framework for ETSC algorithms, the authors measured how different ETSC approaches perform in a RETSC scenario [9]. However, we are not aware of prior work that explicitely targets RETSC.
In this work, we research methods to optimise TEASER for real-time early time series classification while aiming for maximal accuracy.

Realtime

In order to analyse the RETSC problem, it is important to clarify what exactly is meant by “real-time“ in this context. In a more colloquial or non-technical context, “real-time“ typically refers to the ability of a system to respond quickly enough that any delay is imperceptible to a human observer. In technical domains, however, operations may be much faster, and real-time suddenly only allows a delay of less than a few milliseconds [10]. Due to the varying speed requirements of real-time systems across different domains, using minimum response time as a general definition for real-time is not ideal. In technical contexts, therefore, fast computing distinguishes from real-time computing. Rather than being fast, the most important property

that a real-time system should have is predictability, i.e. its functional and timing behaviour should be as deterministic as is necessary to satisfy system specifications [11]. Thus, real-time could be defined as being predictably fast enough to meet the specified constraints.
However, depending on the complexity of a given system, the interpretation of predictability may vary. In simple systems, fulfilment of all specified requirements should always be guaranteed. This is because it is possible to design such systems so that all tasks are fast enough to meet the requirements even in their worst-case runtime. For more complex systems, it is often not possible to accurately determine the worst-case runtime of all components, as the number of tasks, the computational and resource requirements of all tasks, and the environmental influences would have to be known when the system is designed. In this case, predictability is usually not considered as strictly and may vary from task to task. There may be critical tasks for which the fulfilment of their constraints must still be guaranteed. Other tasks may only require a probabilistic guarantee, i.e. only a certain proportion of these tasks must meet their constraints, or the task must meet its constraints with a certain probability [12].
In the field of AI systems, it is common practice to use less strict definitions of real-time, as these systems can be very complex. The focus is usually on achieving high level objectives rather than worst-case requirements. A commonly used definition is that the system is expected to achieve the required quality value in the required time on a statistical basis (e.g. on average). However, no guarantee is given for any individual tasks [13].

TEASER

The Two-tier Early and Accurate Series classifiER (TEASER) was developed as a two-tier approach to the ETSC problem. TEASER works with prefixes of a time series that are extended by a regular number of data points in each step. These prefixes are called snapshots [1]. For each snapshot, a first-tier classifier and an associated second-tier classifier are trained. The first-tier classifier calculates a probability for each class, indicating the likelihood of the current snapshot belonging to that specific class. These class probabilities are forwarded to the second-tier classifier, which then decides whether the first-tier's prediction is reliable enough to return a classification. TEASER does not make a final prediction until a class has been predicted by the second-tier classifier a sufficient number of times in a row. The earliness of this prediction refers to the length of the last snapshot used by the first-tier classifier.
By default, TEASER uses WEASEL[1] [14] as the first-tier classifier and an one-class SVM (ocSVM) as the second-tier classifier. The number of snapshots and therefore also the amount of first-tier and second-tier classifiers, is determined by the interval length parameter set by the user. Typical values for this are 5, 10 or 20, which refer to the number of sections into which the time series data is divided. In each iteration, the snapshots are incremented by one such section. This allows the classifier to avoid processing each new data point individually, which would be less efficient.
In this implementation, real-time does not play a role because new data points are always available as soon as TEASER needs them. Also, the time TEASER needs to classify a data point is irrelevant for achieving a good earliness score. TEASER computes its prediction for a snapshot and moves on to the next longer snapshot as soon as it is complete. In real-time, this may not always be possible due to the gradual arrival of the data.

[1]There is a new Version of WEASEL [15], but TEASER still uses the original Version [14].

Objective

The primary goal of this thesis is to extend TEASER to work in real-time. This means that, on average, TEASER should take less time to classify a data point than it takes to receive a new data point. The average classification time for a data point ($\frac{\sum_{k=1}^{n} t_k}{n}$) must therefore never be longer than one period of the transmission frequency f.

$$\frac{\sum_{k=1}^{n} t_k}{n} \leq \frac{1}{f}$$

In order to be able to meet this real-time hard-constraint, it is first necessary to determine how much time TEASER needs, on average, to classify a data point. Also the transmission frequency needs to be added as a new parameter. This makes it possible to calculate whether the classification is running too fast or too slow in relation to the data rate. TEASER can then adapt its classification to meet the real-time constraint again.
If the transmission frequency is very low, i.e. the data points arrive slowly and TEASER has to wait between classification steps, more time can be spent on classification to hopefully achieve a higher accuracy. This could be done by increasing the amount of new data points in each step and lowering the threshold for how many times a class must be predicted by the second-tier classifier before a final classification is made. If the transmission frequency is very high, i.e. TEASER is not fast enough to process all the incoming data points, then less time should be spent on classification in order to to meet the real-time constraint. In this case, the number of new data points per step could be reduced and the final prediction threshold increased. In addition, new data may be more relevant than old data. Therefore, only every second or third data point could be used for classification. Another approach is to frequently check whether a backlog of data is building up. If this is the case, we can simply discard that data and continue with the latest incoming data point.

Evaluation

To assess the effectiveness of the proposed modifications, data from the UCR Time Series Classification Archive [16], a well known resource in the field of time series classification, will be used. Some of these datasets were also used to evaluate TEASER in the original paper.
TEASER and its extended version will be tested with time series data at different data rates: one lower than TEASER's classification speed, requiring it to wait for the data to arrive; one at the same speed as TEASER's classification speed, for comparison with the basic version; and probably two different data rates higher than TEASER's classification speed, so that a backlog of data will build up. During the experiments, earliness and accuracy are measured. Then, the different approaches to extending TEASER for real-time classification are compared.

References

[1] Schäfer, Patrick and Leser, Ulf (2020) "TEASER: early and accurate time series classification." *Data mining and knowledge discovery, 34(5), 1336-1362.* `https://link.springer.com/article/10.1007/s10618-020-00690-z`

[2] Yang, Xue and Qi, Xuejun and Zhou, Xiaobo (2023) "Deep Learning Technologies for Time Series Anomaly Detection in Healthcare: A Review." *IEEE Access* https://doi.org/10.1109/ACCESS.2023.3325896

[3] Rafiei, Alireza and Zahedifar, Rasoul and Sitaula, Chiranjibi and Marzbanrad, Faezeh (2022) "Automated Detection of Major Depressive Disorder With EEG Signals: A Time Series Classification Using Deep Learning." *IEEE Access, 10, 73804-73817.* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9828387

[4] Majdi I. Radaideh and Connor Pigg and Tomasz Kozlowski and Yujia Deng and Annie Qu (2020) "Neural-based time series forecasting of loss of coolant accidents in nuclear power plants." *Expert Systems with Applications, 160, 113699.* https://doi.org/10.1016/j.eswa.2020.113699

[5] Idrees, Sheikh Mohammad and Alam, M. Afshar and Agarwal, Parul (2019) "A Prediction Approach for Stock Market Volatility Based on Time Series Data." *IEEE Access, 7, 17287-17298.* https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8626097

[6] S. Scarpetta, F. Giudicepietro, E. C. Ezin, S. Petrosino, E. Del Pezzo, M. Martini, M. Marinaro (2005) "Automatic Classification of Seismic Signals at Mt. Vesuvius Volcano, Italy, Using Neural Networks." *Bulletin of the Seismological Society of America, 95(1), 185-196.* https://doi.org/10.1785/0120030075

[7] Gupta, Ashish and Gupta, Hari Prabhat and Biswas, Bhaskar and Dutta, Tanima (2020) "Approaches and Applications of Early Classification of Time Series: A Review." *IEEE Transactions on Artificial Intelligence, 1(1), 47-61.* https://arxiv.org/pdf/2005.02595

[8] Shubhranshu Shekhar, Dhivya Eswaran, Bryan Hooi, Jonathan Elmer, Christos Faloutsos, Leman Akoglu (2023) "Benefit-aware early prediction of health outcomes on multivariate EEG time series" *Journal of biomedical informatics, 139, 104296.* https://doi.org/10.1016/j.jbi.2023.104296

[9] Akasiadis, Charilaos and Kladis, Evgenios and Kamberi, Petro-Foti and Michelioudakis, Evangelos and Alevizos, Elias and Artikis, Alexander (2024) "A Framework to Evaluate Early Time-Series Classification Algorithms" *27th International Conference on Extending Database Technology, 623-635.* https://cer.iit.demokritos.gr/publications/papers/2024/ETSC-EDBT24.pdf

[10] Musliner, D. J., Hendler, J. A., Agrawala, A. K., Durfee, E. H., Strosnider, J. K., Paul, C. J. (1995) "The challenges of real-time AI" *Computer, 28(1), 58-66.* https://www.researchgate.net/publication/2954372_Challenges_of_real-time_AI

[11] Stankovic, J. A. (1988) "Real-time computing systems: The next generation" *University of Massachusetts at Amherst. Computer and Information Science [COINS]* http://se.fi.uncoma.edu.ar/pse2020/apuntes/UM-CS-1988-006.pdf

[12] Shin, K. G., Ramanathan, P. (1994) "Real-time computing: A new discipline of computer science and engineering." *Proceedings of the IEEE, 82(1), 6-24.* https://rtcl.eecs.umich.edu/rtclweb/assets/publications/1994/ramanathan-shin-ieee-proceedings.pdf

[13] Garvey, A., Lesser, V. (1994) "A survey of research in deliberative real-time artificial intelligence." *Real-Time Systems, 6(3), 317-347.* https://web.cs.umass.edu/publication/docs/1993/UM-CS-1993-084.pdf

[14] Schäfer, Patrick and Leser, Ulf (2017) "Fast and Accurate Time Series Classification with WEASEL." *In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 637-646).* https://arxiv.org/pdf/1701.07681

[15] Schäfer, Patrick and Leser, Ulf (2023) "WEASEL 2.0: a random dilated dictionary transform for fast, accurate and memory constrained time series classification." *Machine Learning, 112(12), 4763-4788.* https://doi.org/10.1007/s10994-023-06395-w

[16] Dau, Hoang Anh and Keogh, Eamonn and Kamgar, Kaveh and Yeh, Chin-Chia Michael and Zhu, Yan and Gharghabi, Shaghayegh and Ratanamahatana, Chotirat Ann and Yanping and Hu, Bing and Begum, Nurjahan and Bagnall, Anthony and Mueen, Abdullah and Batista, Gustavo, and Hexagon-ML (2018) "The UCR Time Series Classification Archive." https://www.cs.ucr.edu/~eamonn/time_series_data_2018/