ADecimo: Model Selection for Time Series Anomaly Detection

Paul Boniol  Emmanouil Sylligardos  John Paparrizos  Panos Trahanias  Themis Palpanas
Inria, ENS, CNRS  Inria, ENS, CNRS  The Ohio State University  ICS-FORTH  Université Paris Cité & IUF
boniol.paul@inria.fr emmanouil.sylligardos@ens.fr paparrizos.1@osu.edu trahania@ics.forth.gr themis@mi.parisdescartes.fr

Abstract—Anomaly detection is a fundamental task for time-series analytics with important implications for the downstream performance of many applications. Despite increasing academic interest and the large number of methods proposed in the literature, recent benchmark and evaluation studies demonstrated that there exists no single best anomaly detection method when applied to heterogeneous time series datasets. Therefore, the only scalable and viable solution to solve anomaly detection over very different time series collected from diverse domains is to propose a model selection method that will choose, based on time series characteristics, the best anomaly detection method to run. This paper describes ADecimo, a modular and extensible web application that helps users understand the performance of time series classification algorithms used as model selection methods for time series anomaly detection. Overall, our system enables users to compare 17 different classifiers over 1980 time series, and decide on the most suitable time series classification method for their own time series and use cases.

I. INTRODUCTION

Extensive collections of time-dependent measurements have become a reality in virtually every domain [1]. These measurements result in an ordered sequence of real-valued data points commonly called time series. Managing and analyzing time series collections is becoming important in multiple scientific and industrial applications from various domains, such as astronomy [2], energy sciences [3], and social sciences [4]. Anomaly detection, a particular type of time series analysis, has received ample academic and industrial attention [5], [6], [7], and has become an important problem that finds applications across a wide range of use cases. These applications share the same goal [8], [9], [10]: analyzing time series to identify observations that do not correspond to expected behavior. In practice, anomalies can correspond to [11]: (i) noise or erroneous data (e.g., faulty sensors); or (ii) actual data of interest (e.g., abnormal behavior of the system of interest). In both situations, detecting these anomalies is crucial for many applications.

Several recent surveys and experimental benchmarks describe and study the state-of-the-art time series anomaly detection methods [12], [5], [6], [13], [14]. Interestingly, these benchmark and evaluation studies demonstrated that no single anomaly detection method is the overall best when applied to time series originating from different domains. In practice, we observe that some approaches outperform others on time series with either specific characteristics (e.g., stationary or non-stationary time series), or containing specific types of anomalies (e.g., point-based or sequence-based anomalies).

To overcome the above limitation, ensembling solutions have been proposed [15] that consist of running all existing anomaly detection methods and merging (e.g., averaging) all anomaly scores. Nevertheless, such solutions require executing all methods, thus, leading to running times that are prohibitive for large time series collections.

Therefore, the only scalable and viable solution to solve anomaly detection over time series from heterogeneous sources is to propose a model selection method that will select, based on time series characteristics, the best anomaly detection method to run. The model selection problem for anomaly detection in time series can be formalized as a time series classification problem (i.e., classifying time series into classes corresponding to the correct detectors to select). However, the lack of a benchmark with labeled time series has been a limiting factor for training robust model selection models; this only changed very recently [5], [6], [7].

Based on a very recent experimental evaluation of time series classification methods used as model selection approach for time series anomaly detection [16], we propose ADecimo\(^1\), a system that aims to (i) easily visualize and assess the performances of times series classification methods used as model selection for times series anomaly detection, and (ii) allow the user to use pre-trained model selection method on their own data. ADecimo is based on the TSB-UAD benchmark [5] and employs time series from different domains and applications. Overall, the objective of ADecimo is to (i) facilitate the visualization and the understanding of the performances of model selection applied for time series anomaly detection, (ii) guide users in selecting appropriate approaches for their own use cases, and (iii) promote the development of AutoML methods [17] designed for model selection in the specific context of time series anomaly detection.

II. PRELIMINARIES

We now provide the background necessary for the rest of the paper. We first briefly describe the pipeline used to compare time series classification methods as model selection approaches for time series anomaly detection. We then describe the datasets, anomaly detection methods, time series

\(^1\)Available online: https://adecimots.streamlit.app/
classification approaches, and evaluation measures contained in our system.

A. Evaluation Pipeline

Time series classification can be performed with three strategies: (i) treating the entire time series as one sample, (ii) dividing the time series into overlapping subsequences, and (iii) dividing the time series into shifting subsequences (i.e., non-overlapping subsequences). We consider the third strategy for scalability reasons and to maximize the number of time series classification methods to be considered (not all methods can handle variable length time series).

Therefore, for all time series classification methods, we consider the experimental pipeline, illustrated in Figure 1, with the following steps: (i) Preprocessing step: Extraction of the subsequences of the same length $\ell$ (Figure 1(b)), (ii) Prediction step: Prediction of which detector to use for each subsequence (Figure 1(c)), and (iii) Selection step: Majority voting among all the different prediction to select one detector only (Figure 1(d)). The interested reader can find more technical details in [16].

B. Datasets and Methods

Datasets: We use the public datasets included in the TSB-UAD benchmark [5], comprising 18 datasets used in the literature, for a total of 1980 time series with labeled anomalies. Specifically, each point in every time series is labeled as normal or abnormal. In this demonstration, we consider 16 of the 18 datasets of TSB-UAD (described in Table I).

Anomaly Detection Methods: We select 12 different anomaly detection methods, summarized in Table I. Out of these, 8 are fully unsupervised (i.e., they require no prior information on the anomalies to be detected): IForest, IForest1, LOF, MP, NormA, PCA, HBOS, and POLY. The remaining 4 methods are semi-supervised (i.e., they require some information related to normal behaviors), namely, OCSVVM, AE, LSTM-AD, and CNN. (The parameters for all methods are set as described in the TSB-UAD benchmark [5].)

Method Selection baselines: We then consider the method selection baselines summarized in Table I. We employ feature-based methods, that extract features using the tsfresh [18] library. We also use Rocket, a state-of-the-art time series classifier. Finally, we include two types of deep learning classifiers: (i) Convolutional-based neural networks and (ii) Transformer-based neural networks (all summarized in Table I). In total, we consider 16 methods, trained with windows lengths $\ell$ equal to 16, 32, 64, 128, 256, 512, 768, and 1024, for a total of 128 trained models.

Evaluation measures: We employ 4 evaluation measures. We use classification accuracy for model selection accuracy (i.e., the number of detectors correctly selected divided by the total number of time series). For anomaly detection accuracy, we use AUC-PR [19] and VUS-PR [13] (with a buffer length equal to 10 points). For execution time, we measure the Training time (i.e., the time required to train a model selection method), the Selection time (i.e., the time that a model needs to predict which model to use), and the Detection time (i.e., the time required to predict which detector to use and to run it).
III. ADECIMO: SYSTEM OVERVIEW

In this section, we describe ADecimo\(^2\), the system we developed to help analysts understand the datasets, methods, and results of model selection approaches for time series anomaly detection. The GUI is a stand-alone web application developed using Python 3.6 and the Streamlit framework [20].

Figure 2 illustrates the inputs and features of ADecimo. The system is based on a preloaded set of datasets (16 in our demo), anomaly detection methods (12 in our demo), and accuracy evaluation measures (4 in our demo). The GUI permits interactions with these inputs. First, the user can visualize and interact with the overall experimental evaluation results (by filtering datasets or selecting only a subset of the methods). The user can also visualize the time series, the positions of the anomalies, and the model selection results. Finally, the user can upload their own time series and run our pre-trained model selection approaches.

The GUI is composed of three main frames, shown in Figure 3. The Overall Accuracy and the Overall Execution time frames contain aggregate results (Figure 3(A) for accuracy), the Interactive Exploration can be used to navigate the detailed evaluation results (Figure 3(B)), and the Test on your own data frame allow the user to use the pre-trained model to select a detector for their own time series (Figure 3(C)). In addition, a Description frame contains an overview of the system’s objectives, and a Datasets and Methods frame lists information on each dataset. Finally, the uploaded time series is displayed, and the user can navigate through the time series interactively.

IV. DEMONSTRATION SCENARIOS

This demo has three goals: (i) showcase the importance of a webapp to dive into large experimental results and extract meaningful insight on how much performance a user can gain using model selection on a specific use case; (ii) enable the user to interactively visualize the model selection choices for specific individual time series, models and window length parameter; and (iii) challenge the user to test pre-trained model selection models on a new (or their own) time series.

[Scenario 1: Finding the best model selection method] This scenario starts in frames 1 and 2 (Figure 3(A)). Then, using the sidebar, we will ask the user to select datasets related to their application (e.g., medicine, environmental, or engineering). Then, the user can visit the Datasets frame to get more information on each dataset. Finally, the GUI will depict the most accurate (Figure 3(A)) and most scalable model selection methods (compared to existing anomaly detection methods) interactively. Thus, the user can discover if model selection methods outperform existing anomaly detection methods for a specific application, and if yes, which type of model selection approaches should be used.

[Scenario 2: Understanding and assessing model selection choice] In this scenario, we will ask the user to open the Interactive Exploration Frame (Figure 3(B)). In this frame, the user can select a specific dataset based on their application of interest and a model selection. The user will then be able to select each time series in the chosen datasets and visualize both the time series and the anomaly scores of the selected anomaly detection methods based on the chosen model selection approach. For instance, in Figure 3(B), the user selected one time series from the ECG dataset, and the InceptionTime method with a window length of 16. In this specific example, the user can see that the detector proposed

\(^2\)Available online: https://adecimots.streamlit.app/
by InceptionTime is NormA. As the GUI is also plotting the anomaly scores of all the remaining detectors, the user can assess if the choice made by the chosen model selection approach is correct. We will also explain the behavior of the model selection methods based on the methods’ type and the window length impact.

[Scenario 3: Testing on your own data] In the last scenario, we will ask the user to click on the “upload your own” option in the time series selection drop-down. The user can add a new time series (like in Figure 3(C)). Our system will run the selected model selection approach and the chosen detector. The user will then evaluate the pertinence and accuracy of model selection approaches on the new data.

V. CONCLUSIONS

We demonstrate ADecimo, a system that enables users to understand our experimental evaluation of model selection for time series anomaly detection. It helps users discover if model selection can bring accuracy improvement for specific applications. Moreover, our systems allow the user to understand, assess, and test model selection on each time series of our benchmark and their own data.