Streamify Time-Series Anomaly Detection Methods

M.Sc. Data and Web Science – DWS203: Mining of Massive Datasets

Introduction

Through this project, the goal is to study streaming versions of anomaly detection methods for time series. We have discussed that time series are ubiquitous across scientific domains and industrial settings. From the time-series analytical tasks, anomaly detection stands out due to the difficulty in the detection of outliers as well as the importance of the task in real-world applications. The aim is to gain experience working with 1) the state-of-the-art benchmarks, methods, and evaluation measures in the area and 2) study how different anomaly detection methods can be modified (adapted) to work on streaming scenarios where data arrive progressively.

Steps:

1) Read the SAND paper

   Paper: https://www.paparrizos.org/papers/BoniolVLDB21a.pdf

   Code: https://github.com/TheDatumOrg/TSB-UAD/blob/main/TSB_UAD/models/sand.py

2) TSB-UAD is a recent benchmark for anomaly detection with about 2000 labeled examples from different domains. SAND relies on some of them in its evaluation

   Paper: https://www.paparrizos.org/papers/PaparrizosVLDB22a.pdf

   Code: https://github.com/TheDatumOrg/TSB-UAD

   Public dataset: https://www.thedatum.org/datasets/TSB-UAD-Public.zip

3) Pick a few time series from different domains in the Public dataset of TSB and concatenate them (ts1, ts2, ..) to replicate the streaming settings in the SAND paper (we evaluate the method using different notions of normality. Normality 1 consists of a single time series (no distribution shift), Normality 2 consists of two concatenated time series from different domains/datasets (1 distribution shift), Normality 3 consists of three concatenated time series from different domains/datasets (2 distribution shifts), etc.

Now the goal is to evaluate offline (non-streaming) and streaming methods on the generated datasets of step 3. We move forward as follows:

4) Pick two non-streaming methods of your choice as a baseline (e.g., Isolation Forest and a DNN method). Run the methods on the generated datasets (no streaming setting; use the entire generated datasets as input). These methods will serve as our offline baselines.

5) Run the SAND method. This will be your STREAMING baseline. Evaluate the performance in “online” settings (data arrive in batches)

6) Modify the chosen methods (Step4) to operate in streaming settings. You should implement at least 2 streaming variants.

   a. Offline baseline: Run the methods using the entire dataset as input (offline setting – we see the future immediately)
   b. Streaming baseline: Run SAND using the “online” version – data arrive incrementally
   c. Variant 1. Apply your chosen methods to each batch of the data that arrives. No modification of the method itself. (Naïve streaming variant – see NormA-Batch or S2G-Batch examples in SAND paper)
   d. Variant 2. Suggest modifications/adaptations to improve the accuracy (and likely runtime) performance compared to Variant 1. Different modifications per method are also OK.

   (choose methods from different categories, let’s say one unsupervised one semi-supervised, or one from traditional literature one from the modern DNN methods)

Deliverables

- Your code (and if possible the Jupyter Notebook in PDF/HTML format)
- Your generated datasets with different levels of normality
- A presentation 5-10 slides, explaining your methods, datasets, and modification (variant 2). The slides should include accuracy/runtime plots/tables as in SAND to demonstrate the performance of your streaming variants. Please note that you are going to present your work in class, most probably during the last lecture of the semester