Evaluation of anomaly detection methods in time series using Instance Space Analysis

Ilyas Lemmens[1,2], Louis Carpentier[1,2], Wannes Meert[1], and Mathias Verbeke[1,2]

[1] KU Leuven, Department of Computer Science, DTAI, B-3000 Leuven, Belgium
[2] Flanders Make@KU Leuven, Belgium
{firstname.lastname}@kuleuven.be

Time series anomaly detection (TSAD) refers to identifying unusual patterns in sequential time series data. Detecting anomalous data is crucial, as anomalies often indicate critical events or problems within the monitored system. Numerous algorithms have been developed for this task, each with its own assumptions. Recent research, such as the extensive evaluation by Schmidl et al. (2022), has shown that there is no single algorithm that universally outperforms all others. This aligns with Wolpert and Macready's (1997) *No Free Lunch Theorem*, which asserts that no single algorithm is best for every problem.

This thesis abstract summarizes the main findings of Lemmens (2024). The main goal of this work is to analyze the performance of TSAD algorithms and answer the research question: "Which characteristics of time series indicate the success or failure of certain (classes of) methods for anomaly detection in time series?". To achieve this, we employ Instance Space Analysis (ISA), a method introduced by Smith-Miles et al. (2014) that allows for comprehensive evaluation by comparing algorithms in a reduced 2D instance space with the following steps:

1. Collect useful problem instances, algorithms and a performance metric. Calculate descriptive meta-features (e.g., minimum value) for the instances.
2. Project time series instances into a 2D instance space based on their descriptive features and algorithm performance to reveal significant trends.
3. Generate footprints: predictive areas in the instance space that indicate where an algorithm is likely to perform well.
4. Generate additional figures for further analysis. Such as feature distributions and areas where algorithms are expected to outperform the other algorithms.

For this study, we used MATILDA, an advanced implementation of ISA developed by Smith-Miles and Muñoz (2023), to analyze 1525 benchmark time series datasets. We evaluated 10 popular TSAD algorithms using 33 different features and multiple performance metrics, including VUS-PR as proposed by Paparrizos et al. (2022).

Our analysis yielded several useful insights. Figure 1 illustrates the instance space with the instances colored according to their original dataset. We demonstrate that instances from the same dataset often tend to cluster together, indicating that they share similar characteristics. Consequently, it is essential to use multiple diverse datasets to fairly evaluate TSAD algorithms. The algorithmic footprints, shown in Figure 2, reinforce the notion that no single algorithm excels across all instances. Furthermore, the 10 algorithms are classified into 4 different groups, each performing better in different areas of the instance space. By cross-referencing the algorithmic footprints with the distribution of our features across the instance space, we were able to deduce that the performance of an anomaly detector seems to correlate with features such as skewness, kurtosis and absolute energy.

Figure 2 was used to tackle Rice's Algorithm Selection Problem, which aims to select the best algorithm for a new problem instance. The figure indicates the complexity in deducing the very best algorithm for an instance. The analysis revealed that there is a clear portion of the instance space where the SAND algorithm consistently outperforms the others. Yet, outside of this area, algorithm selection is still fragmented and dubious.

To further explore the effectiveness of a meta-algorithm for algorithm selection, we developed and evaluated a meta-algorithm based on our insights from the ISA. The meta-algorithm combines Manifold Learning with a classifier and was assessed using stratified 10-fold cross-validation. Our results demonstrate that the meta-algorithm significantly enhances performance compared to applying any individual algorithm to all instances. The best performing meta-algorithm, combining t-SNE and Random Forest, achieved an average VUS-PR score of 0.468 and a classification accuracy of 46.88%. Applying SAND, the detector with highest average performance, to all instances results in a VUS-PR score of 0.376 and classification accuracy 37.4%. These findings suggest a promising start for further improving TSAD performance through adaptive meta-algorithms.

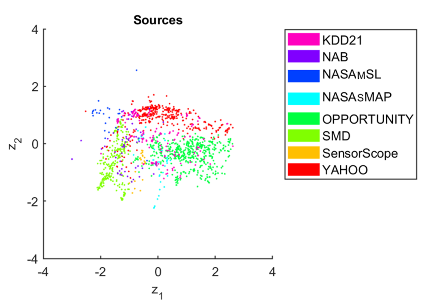

Fig. 1. Distribution of datasets across the Instance Space.

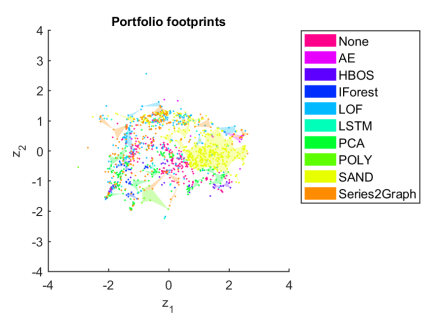

Fig. 2. Portfolio of footprints that demonstrate which algorithm is likely to perform best.

Acknowledgements

This research is supported by the Flemish government under the Flanders AI Research Program, by Internal Funds KU Leuven (STG/21/057), and the Flanders Make Strategic Basic Research project SAFOS.

References

1. Ilyas Lemmens. Evaluation of anomaly detection methods in time series using Instance Space Analysis. Master's thesis, KU Leuven, 2024.
2. Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. Proc. VLDB Endow., 15(9):1779–1797, 2022.
3. D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1):67–82, 1997.
4. Kate Smith-Miles, Davaatseren Baatar, Brendan Wreford, and Rhyd Lewis. Towards objective measures of algorithm performance across instance space. Computers & Operations Research, 45:12–24, 2014.
5. John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. Proc. VLDB Endow., 15(11):2774–2787, 2022.
6. Kate Smith-Miles and Mario Andrés Muñoz. Instance space analysis for algorithm & nbsp;testing: Methodology and software tools. ACM Comput. Surv., 55(12), 2023.
7. John R. Rice. The algorithm selection problem. this chapter was presented as the george e. forsythe memorial lecture at the computer science conference, February 19, 1975, Washington, D. C. volume 15 of Advances in Computers, pages 65–118. Elsevier, 1976.