Unveiling the Flaws: A Critical Analysis of Initialization Effect on Time Series Anomaly Detection

Alexander Koran [1,2] Hadi Hojjati [1,2], Narges Armanfard[1,2]
[1]Department of Electrical and Computer Engineering, McGill University
[2]MILA - Quebec AI Institute
Montreal, QC, Canada
narges.armanfard@mcgill.ca

Abstract

Deep learning for time-series anomaly detection (TSAD) has gained significant attention over the past decade. Despite the reported improvements in several papers, the practical application of these models remains limited. Recent studies have cast doubt on these models, attributing their results to flawed evaluation techniques. However, the impact of initialization has largely been overlooked. This paper provides a critical analysis of the initialization effects on TSAD model performance. Our extensive experiments reveal that TSAD models are highly sensitive to hyperparameters such as window size, seed number, and normalization. This sensitivity often leads to significant variability in performance, which can be exploited to artificially inflate the reported efficacy of these models. We demonstrate that even minor changes in initialization parameters can result in performance variations that overshadow the claimed improvements from novel model architectures. Our findings highlight the need for rigorous evaluation protocols and transparent reporting of preprocessing steps to ensure the reliability and fairness of anomaly detection methods. This paper calls for a more cautious interpretation of TSAD advancements and encourages the development of more robust and transparent evaluation practices to advance the field and its practical applications.

Introduction

Time series anomaly detection (TSAD) has been a significant area of interest in the field of data science due to its significant applications across various domains, including transportation (Hojjati, Sadeghi, and Armanfard 2023), healthcare (Ho and Armanfard 2023), and industrial monitoring (Hojjati and Armanfard 2022). The recent surge in interest is partly driven by the advancements in deep learning, which have shown promising results in handling time series data for tasks such as classification and forecasting (Torres et al. 2021).

In recent years, most research in time-series anomaly detection focused on modifying visual anomaly detection frameworks by incorporating a customized temporal architecture to handle sequential data. Early experiments have hinted that these algorithms can significantly outperform the state-of-the-art (Choi et al. 2021). However, recent studies have cast a shadow of doubt over these claims (Garg et al. 2022). They have shown that the performance of deep time-series models is not necessarily better than traditional time-series anomaly detection algorithms or even, in some cases, random baselines (Kim et al. 2022). The superior performance that some papers have reported can be attributed to other factors, such as incorrect evaluation procedures and inappropriate benchmark detests (Wu and Keogh 2023). As a result, most of the recent progress in deep time-series anomaly detection is practically useless in many real-life applications.

Initialization is one of the key aspects of time-series analysis that potentially plays an important role in the performance and reliability of TSAD algorithms. It can lead to misleading conclusions about the efficacy of TSAD algorithms, as the initialization steps can significantly affect the anomaly detection results. This phenomenon challenges the reliability of a TSAD method, as it makes it difficult to distinguish between genuine performance improvements due to the model and variations in results induced by initialization.

Despite the importance of initialization, its impact on anomaly detection performance has not been thoroughly investigated. In this paper, we provide a critical analysis of how initialization affects TSAD performance. We aim to uncover the flaws and biases introduced by commonly employed initialization and preprocessing techniques. Through extensive experiments, we demonstrate how different initialization techniques can significantly alter detection outcomes, highlighting the risk of artificial performance boosts resulting from variations in initialization.

Our findings suggest that while initialization is an integral part of time series processing, it must be applied judiciously, with a clear understanding of its potential impact on anomaly detection results. By shedding light on the complexities and pitfalls of initialization in time series anomaly detection, we hope to contribute to the development of more robust and reliable detection methodologies, ultimately advancing the field and its practical applications.

Background

Anomaly Types

The primary goal of TSAD is to detect anomalies that deviate from the expected data patterns, which can be classified into three main types (Hojjati, Ho, and Armanfard 2024):

Contextual Anomalies: These anomalies occur when data points significantly differ from the normal pattern based on the context, such as seasonal trends or temporal dependencies. For instance, unusually high electricity usage during a typically low-demand period could be a contextual anomaly.

Collective Anomalies: These involve a series of data points that, when considered together, deviate from the normal pattern. An example could be a sudden spike in network traffic that, while individual data points may not seem anomalous, collectively indicate a potential security breach.

Point Anomalies: These are individual data points that significantly deviate from the expected value. Point anomalies are very common in current TSAD datasets and can include instances like a sudden drop in stock prices.

Time-Series Anomaly Detection Methods

TSAD has been a longstanding research problem. Several types of methods have been developed to tackle it in the past literature:

Statistical Methods: These include techniques like Z-score, moving averages, and Holt-Winters exponential smoothing (Munir et al. 2019). These methods are straightforward and computationally inexpensive but often lack robustness against complex and high-dimensional data. They assume that data follows a specific statistical distribution, which is not always the case in real-world scenarios.

Traditional Machine Learning Methods: Unsupervised methods such as Isolation Forest (Liu, Ting, and Zhou 2008), One-Class SVM and SVDD (Hojjati and Armanfard 2023), and clustering-based approaches (Sadeghi et al. 2023). These methods learn the normal behaviour from training data and identify deviations in test data. They do not require labelled data, making them suitable for real-world applications where anomalies are rare and difficult to label.

Deep Learning Methods: Techniques like Autoencoders, LSTM networks, and Generative Adversarial Networks (GANs) have shown great promise in TSAD (Choi et al. 2021). These methods can model complex temporal dependencies and detect anomalies based on reconstruction errors (e.g., Autoencoders) or prediction deviations (e.g., LSTMs). They require significant computational resources and are often data-hungry, but their ability to capture non-linear relationships in the data makes them powerful tools for TSAD.

Challenges in Time Series Anomaly Detection

Despite the recent advancements in TSAD, several issues remain unresolved, leading to challenges in accurately evaluating and comparing different anomaly detection methods:

Evaluation Metrics: Traditional point-based metrics (e.g., precision, recall, F1-score) often fall short of accurately reflecting the performance of TSAD methods, especially for range-based anomalies. Most anomalies in time series data appear consecutively, forming anomaly segments. Point-based evaluation metrics evaluate each observation independently, which can lead to considerable underestimation or overestimation of detection performance.

Recent studies have proposed adjusted evaluation protocols to address these limitations. Point Adjustment (PA) (Liu et al. 2023) is one such protocol where all points within an anomaly segment are considered true positives if any point within the segment is detected. However, PA can significantly overestimate performance and create an illusion of superior detection capability. Variants like PA%K (Kim et al. 2022) apply PA only if the ratio of correctly detected anomalies to the segment length exceeds a threshold, aiming to balance precision and recall more effectively. Despite these efforts, the search for a universally effective evaluation metric continues.

Benchmark Datasets: Commonly used benchmark datasets for TSAD, such as Yahoo (Laptev, Amizadeh, and Billawala 2015) and Numenta (Ahmad et al. 2017), have been found to have several flaws, including being too simplistic or not representing the complexities of real-world scenarios (Wu and Keogh 2023). These datasets often contain artificially generated anomalies, which do not capture the subtlety and diversity of anomalies encountered in practice. Consequently, methods that perform well on these benchmarks may fail when applied to real-world data, leading to an illusion of progress in the field of TSAD. There is a need for more comprehensive and challenging datasets that better represent real-world conditions to truly evaluate and advance TSAD methods.

Model Interpretability: Deep learning models, while powerful, often operate as black boxes, making it challenging to interpret their predictions. Understanding why a model flagged certain data points as anomalies is crucial for trust and adoption in critical applications. Methods like LIME and SHAP (A. and R. 2023) have been developed to provide explanations for model predictions, but these techniques are still in their infancy for time series data and need further refinement to be effectively applied in TSAD.

Initialization Effect: Initialization steps such as preprocessing, seed variability, and window size selection are crucial for Time Series Anomaly Detection (TSAD). However, these steps can potentially introduce biases that affect anomaly detection performance. Some initialization steps might create shortcuts for the model, compromising the robustness of the detection process. Therefore, understanding the potential effects of each preprocessing step is essential to ensure the reliability of TSAD methods. Without this understanding, a method might falsely claim to improve state-of-the-art (SOTA) results by inflating its performance through choosing a specific seed or hyperparameters.

Our Contribution

By unveiling the flaws and biases introduced by common initialization techniques, we seek to provide a more nuanced understanding of their effects on the accuracy and reliability of anomaly detection methods. Through extensive experiments on benchmark datasets, we aim to demonstrate how different initialization steps can significantly alter detection outcomes. In summary, our contributions are as follows:

1. We evaluate various initialization techniques and their impact on the performance of TSAD methods.
2. We demonstrate how significantly the choice of hyperparameters can impact performance. In many cases, the variation of results caused by altering a benign hyperparameter such as the seed number is far greater than the claimed improvements over the SOTA.
3. By shedding light on this overlooked area of TSAD, we aim to encourage researchers to critically analyze the results of previous work and conduct experiments in future method developments to ensure that performance improvements are due to the model, not the choice of initialization parameters.

Experiment Setup

In order to demonstrate the effect of various initialization techniques on the performance of TSAD methods, we designed a comprehensive experimental setup. This involved systematically varying key initialization parameters on benchmark datasets.

Datasets

Secure Water Treatment (SWaT) Dataset (Goh et al. 2016): The SWaT dataset is derived from a water treatment testbed built at the Singapore University of Technology and Design. It is designed to simulate a real-world water treatment process and contains data that represent normal operations as well as various attack scenarios that cause anomalies. The dataset includes a comprehensive range of sensor readings and actuator statuses collected over a period of days. SWAT is considered to be a standard benchmark, yet similar to other TSAD datasets, SWAT suffers from issues such as labelling discrepancy.

Server Machine Dataset (SMD) (Su et al. 2019): The SMD dataset consists of multivariate time-series data collected from 28 server machines in a large internet company over five weeks. Each server is monitored by 33 sensors, capturing metrics such as CPU load, network usage, and memory usage. This dataset is particularly valuable due to its scale and the diversity of operational conditions it encompasses. Following (Lai, Ho, and Armanfard 2024), in our study, we modified the dataset and used an interval-segmented version of SMD. This approach increases the complexity of the dataset and helps mitigate the issue of triviality.

NIPS-TS-GECCO (Moritz et al. 2020): The GECCO dataset is derived from water quality data collected by Thüringer Fernwasserversorgung, a German water supply company operating an extensive network of reservoirs and treatment plants. The dataset was created for the GECCO 2018 Industrial Challenge, which aimed to develop methods for online monitoring and change detection in time series data of drinking water composition. It contains real-world and simulated water quality indicators, such as pH, chlorine dioxide levels, and turbidity, annotated with events that represent significant changes.

Algorithms

For our experiment, we used three state-of-the-art (SOTA) methods that have been commonly employed by TSAD researchers for comparison and benchmarking over the past few years.

Graph Deviation Network (GDN): GDN (Deng and Hooi 2021) is designed to detect anomalies in multivariate time series data by leveraging the inherent structure of the data. The idea behind GDN is to model the relationships between different time series using a graph structure where each node represents a time series, and edges capture the dependencies between them. GDN uses Graph Neural Networks (GNNs) to learn these dependencies and identify deviations from normal behavior. In GDN, each node corresponds to a sensor, and the edges are weighted based on the correlation between the sensors. The network is trained to predict the value of each sensor at a given time step based on the values of the other sensors and itself in the past. Anomalies are detected when the actual sensor values significantly deviate from the predicted values. This method allows GDN to capture and leverage the spatial dependencies in multivariate time series data.

Multivariate Time-Series Anomaly Detection via Graph Attention Networks (MTAD-GAT): MTAD-GAT (Zhao et al. 2020) employs a combination of convolutional layers, graph attention networks, and recurrent units to capture both spatial and temporal dependencies in multivariate time series data. The architecture begins with a 1-D convolutional layer that extracts high-level features from each time series, enabling the network to understand the inherent patterns within the data. Following this, two parallel Graph Attention Networks (GAT) layers are utilized: one focusing on feature-oriented dependencies and the other on time-oriented dependencies. This dual approach enables the model to capture the intricate relationships between different features and across different time steps. The outputs from the convolutional and GAT layers are then fed into a Gated Recurrent Unit (GRU), which models the sequential patterns in the data, providing a deeper understanding of temporal dependencies. MTAD-GAT uses both a forecasting model, which predicts the next time step, and a reconstruction model, which employs a Variational Autoencoder (VAE) to learn a latent representation of the time series.

Unsupervised Anomaly Detection (USAD): USAD (Audibert et al. 2020) is an unsupervised anomaly detection method for multivariate time series, utilizing a combination of autoencoders and adversarial training. Initially, the method involves training an autoencoder to minimize reconstruction errors, which helps learn the normal patterns present in the data. Following this, USAD incorporates adversarial training, where the objective is to differentiate between the actual time series data and the reconstructed data generated by the autoencoder. This adversarial phase aims to

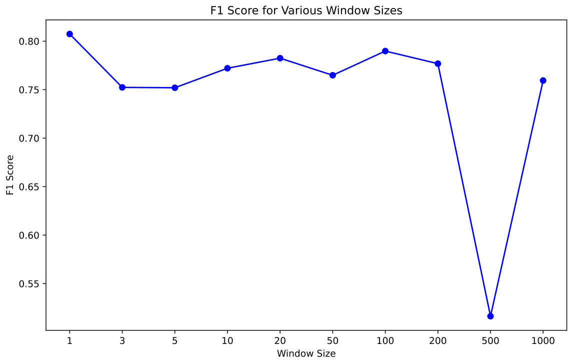


Figure 1: F1 scores obtained by GDN on the SWAT dataset for varying window sizes.

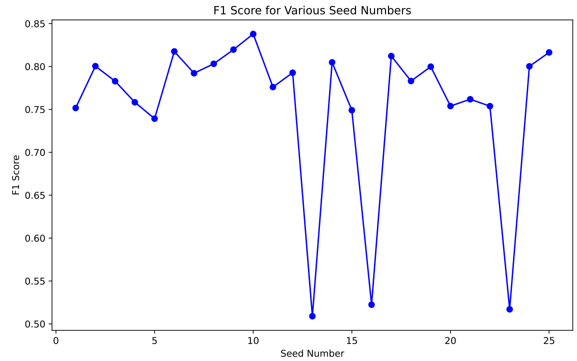


Figure 2: F1 scores obtained by GDN on the SWAT dataset for varying random seed numbers.

amplify the reconstruction error when anomalies are present, thus making the model more sensitive to deviations.

Results and Discussion

Window Size Effect

Window size is a critical parameter in TSAD as it defines the context for anomaly detection. Choosing an appropriate window size is essential because it influences the model's ability to capture temporal dependencies.

A small window size captures fine-grained temporal patterns but may miss broader trends and context. Conversely, a larger window size captures broader temporal trends but may smooth over important short-term anomalies. The choice of window size should balance the need to capture relevant patterns while avoiding the dilution of critical anomalies. Ideally, a reliable method should not be hypersensitive to this parameter. While the previous discussion is valid for very large or very small window sizes, minor adjustments in window size should not significantly impact performance in an ideal scenario. To assess the effect of window size on the anomaly detection performance of our TSAD methods, we set it to 1, 3, 5, 10, 20, 50, 100, 200, 500, and 1000 to represent a wide range of possible values. The resulting figure is plotted in Figure 1.

In this figure, we observe the F1 scores for various window sizes when running GDN on the SWAT dataset. The variability in the model's performance across different window sizes is immediately noticeable. For instance, changing the window size from 1 to 3 results in more than a 5% decrease in F1 score. This should not occur; in fact, we should expect the model to perform better when provided with more data within the window. It is also worth noting that even a 3% improvement is considered groundbreaking in the literature. The sharp drop in performance for a window size of 500 is also alarming. Although several trials suggest it may be an outlier, this is still far from ideal, as an anomaly detection model in practice should not have drastic dips in performance merely becasue of changing a parameter.

Not only is there a lack of correlation between window size and performance, but using a window size of 1 actually yielded the highest F1 score in this experiment. This is unexpected because GDN is a predictive method, and a window size of 1 means the model only analyzes the previous time point when determining if a certain time point is anomalous. This suggests that GDN on SWAT is not effectively capturing long-term dependencies when using larger window sizes; if it were, we would expect better performance for window sizes above 1, at least up to a certain point. We performed similar window size experiments on the MTAD-GAT and USAD models and observed the same trend, as discussed in the Supplementary Material section.

Our main takeaway is that these models do not sufficiently utilize their window size architecture to capture long-term dependencies in datasets such as SWAT. Additionally, they suffer from large performance variance relative to the window size, often much more than 5%. This variation in performance is far greater than what current methods achieve over traditional methods, challenging the reliability of their claims, as the observed performance boost might be attributed to changing a parameter such as window size. It is also worth noting that there is no consistency in the literature regarding the standard window size for a dataset. Furthermore, some papers claim to downsample the data due to memory concerns, which also impacts the amount of information in each window and can drastically affect performance compared to the existing SOTA methods.

Seed Variability

Many model initialization and data preprocessing steps involve randomness. To ensure the reproducibility and reliability of results, it is crucial to evaluate the impact of seed variability. Running experiments with different seeds helps in understanding the stability of the initialization methods and the algorithms' performance. Ideally, changing the seed number when running a method should not significantly affect performance. If it does, it suggests that the method is not reliable and that optimal seed selection might have been done using grid search to inflate the model's performance compared to established baselines.

In our experiments, we ran the models with their default

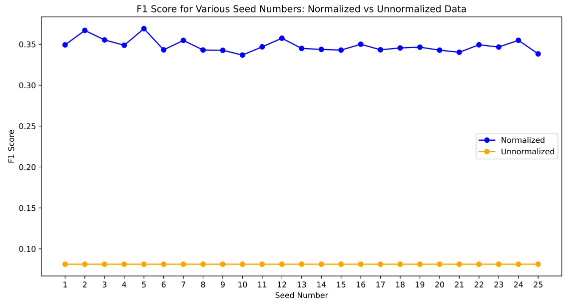


Figure 3: F1 scores obtained by USAD on the SWAT dataset for varying random seed numbers. The blue line is the USAD implementation which has normalization, whereas the orange line is USAD without using normalization.

parameters but fixed the seed to various seed numbers. We chose seed numbers $\{1, 2, \ldots, 25\}$ in these experiments, although any 25 different numbers would work. The goal was to examine the variability of the models.

Figure 2 illustrates 25 trials of GDN on SWAT with the exact same initial parameters. Among the first 10 seeds, there are sharp increases and decreases, many exceeding 5% changes or more. This already supports our claim from the previous section that the model's results are too variable and the models are hypersensitive to parameters, which should not be the case.

Upon inspecting and plotting 25 trials, we observe even more drastic changes in performance. In 3 of these 25 seeds, the model's performance dropped below a 0.55 F1 score, representing more than a 25% change from the other F1 values. This is very alarming, as we would expect a reliable model to be robust across multiple trials with the same input parameters.

Normalization

Normalization is the process of scaling data to a fixed range, typically between 0 and 1, or to have a mean of 0 and a standard deviation of 1. This step is crucial for ensuring that different features of time series data contribute equally to the anomaly detection process. Without normalization, features with larger numerical ranges can disproportionately influence the model's performance, leading to biased results. Proper normalization ensures that all features contribute equally to the anomaly detection task, improving the model's accuracy and efficiency. However, inappropriate normalization can smooth out anomalies or fail to capture critical variations in the data, leading to decreased detection performance.

In our experiments, we ran USAD on the SWAT dataset, both with and without normalization, across 25 different seeds. Figure 3 illustrates the F1 scores obtained from these trials. The results show that normalization significantly impacts the performance and variability of the USAD model. With normalization, the model achieved far higher F1 scores, indicating vast improvement in performance. However, the model also exhibited greater variability, suggesting that normalization might introduce sensitivity to the initial parameters.

These results also hint that the model is extremely sensitive to normalization. This sensitivity can be exploited to artificially inflate the performance of an TSAD method. By selecting specific normalization techniques or tuning parameters to favor the model, one can achieve seemingly superior results that may not generalize well to other datasets or real-world applications. The inconsistency in performance due to normalization underscores the need for rigorous evaluation and transparent reporting of the input data's characteristics and preprocessing steps to ensure the reliability and fairness of anomaly detection methods.

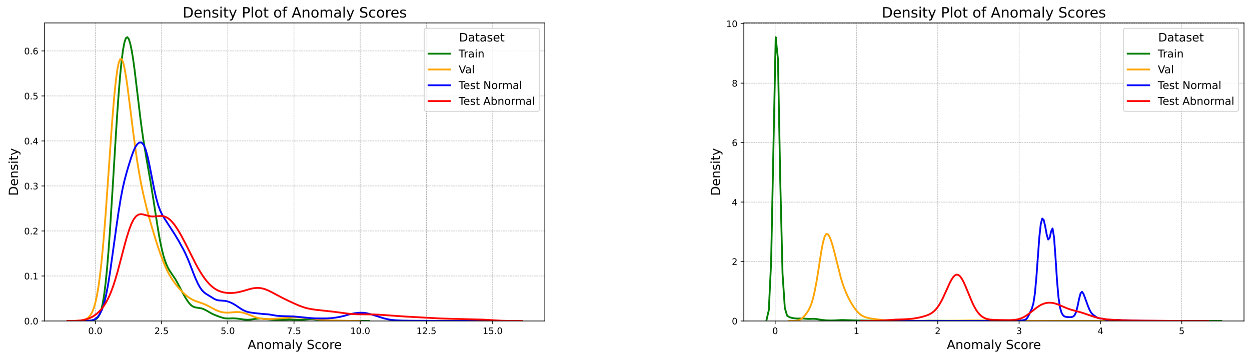
Analysis of Anomaly Score

As observed in previous sections, many seemingly unrelated parameters can significantly affect the performance of a TSAD model. As several studies have hinted earlier, the current numerical improvements reported by many papers are often owed to flawed thresholding and evaluation protocols . To have a more realistic view of the true output of TSAD models under different conditions, we analyze the density of raw anomaly scores before applying any transformation, thresholding, or point adjustment. We plot the density for training, validation, and test sets: in the SWAT and SMD datasets, the training and validation sets consist only of normal data, while the test set includes both normal and abnormal data. Ideally, we would like to observe significantly low anomaly scores for the training, validation, and normal test data, and significantly higher anomaly scores for the abnormal test data.

Figure **??** illustrates the anomaly score density of GDN on the SMD dataset. Densities of other methods and datasets are also plotted in the supplementary materials. In this figure, we can observe that the anomaly scores of the training samples are what we expect: near zero, since the model learned during training that these were normal. The validation set, which consists of unseen normal data, also shows anomaly scores close to zero.

However, the density of the anomaly scores for abnormal samples during the test phase does not significantly differ from the normal data, and is still close to zero. This explains why the model does not perform well in this experimental setup: it cannot distinguish between normal and abnormal samples. Yet, by incorporating schemes such as point adjustment, it might be possible to report an acceptable F1 score.

Another interesting and extreme example of counterintuitive anomaly score densities is observed with MTAD-GAT on the SWAT dataset. As Figure **??** demonstrates, the anomaly scores of the normal test data are even larger than those of the abnormal data. This is counterintuitive given that the model is trained to minimize the anomaly score on normal data. However, since the density of normal test data scores does not fully overlap with the distribution of abnormal anamaly scores, the model could still achieve a good F1 score by using a kernel to transform the anomaly scores and thresholding to isolate the densities. This highlights how



(a) GDN on the SMD dataset.

(b) MTAD-GAT on the SWAT dataset.

Figure 4: Density plots of anomaly scores across training, validation, normal test, and abnormal test sets. (a) GDN on SMD. (b) MTAD-GAT on SWAT.

post-processing techniques can mask underlying model deficiencies and produce seemingly favorable results.

At the cost of sacrificing the robustness of the model, one could thoroughly search for a set of optimal initialization parameters such as window size, seed number, and normalization technique. Combined with existing flawed evaluation protocols, this approach could potentially outperform any SOTA method without actually improving the model's impact on real-world TSAD problems. This underscores the need for rigorous evaluation and transparent reporting of preprocessing steps and input data characteristics to ensure the reliability and fairness of anomaly detection methods.

Conclusion

This paper critically analyzed the impact of initialization on Time Series Anomaly Detection (TSAD) models. Our experiments revealed that these models are highly sensitive to parameters like window size, seed number, and normalization, leading to significant performance variability. This sensitivity can be exploited to artificially inflate performance results, raising concerns about the robustness and reliability of current TSAD methods. Our findings highlight the necessity for rigorous evaluation protocols and transparent reporting of preprocessing steps to ensure the reliability and fairness of anomaly detection methods.

To advance the field, we recommend that future research focuses on developing more robust models that are less sensitive to initialization parameters. Additionally, standardizing evaluation protocols and ensuring transparency in reporting preprocessing steps are essential to prevent the exploitation of these sensitivities. By addressing these issues, we can develop more reliable and fair anomaly detection methods that are truly beneficial in real-world applications.

References

A., S.; and R., S. 2023. A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends. *Decision Analytics Journal*, 7: 100230.

Ahmad, S.; Lavin, A.; Purdy, S.; and Agha, Z. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262: 134–147. Online Real-Time Learning Strategies for Data Streams.

Audibert, J.; Michiardi, P.; Guyard, F.; Marti, S.; and Zuluaga, M. A. 2020. USAD: UnSupervised Anomaly Detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3395–3404. Virtual Event, CA, USA: ACM.

Choi, K.; Yi, J.; Park, C.; and Yoon, S. 2021. Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE access*, 9: 120043–120065.

Deng, A.; and Hooi, B. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4027–4035.

Garg, A.; Zhang, W.; Samaran, J.; Savitha, R.; and Foo, C.-S. 2022. An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6): 2508–2517.

Goh, J.; Adepu, S.; Junejo, K. N.; and Mathur, A. 2016. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In *The 11th International Conference on Critical Information Infrastructures Security*.

Ho, T. K. K.; and Armanfard, N. 2023. Self-Supervised Learning for Anomalous Channel Detection in EEG Graphs: Application to Seizure Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7): 7866–7874.

Hojjati, H.; and Armanfard, N. 2022. Self-Supervised Acoustic Anomaly Detection Via Contrastive Learning. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3253–3257.

Hojjati, H.; and Armanfard, N. 2023. DASVDD: Deep Autoencoding Support Vector Data Descriptor for Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering*, 1–12.

Hojjati, H.; Ho, T. K. K.; and Armanfard, N. 2024. Self-supervised anomaly detection in computer vision and beyond: A survey and outlook. *Neural Networks*, 172: 106106.

Hojjati, H.; Sadeghi, M.; and Armanfard, N. 2023. Multivariate Time-Series Anomaly Detection with Temporal Self-supervision and Graphs: Application to Vehicle Failure Prediction. In De Francisci Morales, G.; Perlich, C.; Ruchansky, N.; Kourtellis, N.; Baralis, E.; and Bonchi, F., eds., *Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track*, 242–259. Cham: Springer Nature Switzerland. ISBN 978-3-031-43430-3.

Kim, S.; Choi, K.; Choi, H.-S.; Lee, B.; and Yoon, S. 2022. Towards a Rigorous Evaluation of Time-Series Anomaly Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7): 7194–7201.

Lai, T.; Ho, T. K. K.; and Armanfard, N. 2024. Open-Set Multivariate Time-Series Anomaly Detection. In *Proceedings of the 27th European Conference on Artificial Intelligence*.

Laptev, N.; Amizadeh, S.; and Billawala, Y. 2015. S5 A Labeled Anomaly Detection Dataset, Version 1.0 (16M).

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*, 413–422.

Liu, M.; Wang, Y.; Xu, H.; Zhou, X.; Li, B.; and Wang, Y. 2023. Smoothing Point Adjustment-Based Evaluation of Time Series Anomaly Detection. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

Moritz, S.; Rehbach, F.; Chandrasekaran, S.; Rebolledo, M.; and Bartz-Beielstein, T. 2020. GECCO Industrial Challenge 2018 Dataset: A water quality dataset for the 'Internet of Things: Online Anomaly Detection for Drinking Water Quality' competition at the Genetic and Evolutionary Computation Conference 2018, Kyoto, Japan.

Munir, M.; Chattha, M. A.; Dengel, A.; and Ahmed, S. 2019. A Comparative Analysis of Traditional and Deep Learning-Based Anomaly Detection Methods for Streaming Data. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 561–566.

Sadeghi, M.; Hojjati, H.; Armanfard, N.; and Institute, M. Q. A. 2023. C3: Cross-instance guided Contrastive Clustering. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA.

Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; and Pei, D. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In Teredesai, A.; Kumar, V.; Li, Y.; Rosales, R.; Terzi, E.; and Karypis, G., eds., *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2828–2837.

Torres, J. F.; Hadjout, D.; Sebaa, A.; Martínez-Álvarez, F.; and Troncoso, A. 2021. Deep Learning for Time Series Forecasting: A Survey. *Big Data*, 9(1): 3–21. PMID: 33275484.

Wu, R.; and Keogh, E. J. 2023. Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress. *IEEE Transactions on Knowledge and Data Engineering*, 35(3): 2421–2429.

Zhao, H.; Wang, Y.; Duan, J.; Huang, C.; Cao, D.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; and Zhang, Q. 2020. Multivariate Time-Series Anomaly Detection via Graph Attention Network. In *2020 IEEE International Conference on Data Mining (ICDM)*, 841–850. Los Alamitos, CA, USA: IEEE Computer Society.