GNNs for Time Series Anomaly Detection: An Open-Source Framework and a Critical Evaluation

Federico Bello[1], Gonzalo Chiarlone[12], Marcelo Fiori[13][a], Gastón García González[1][b] and Federico Larroca[13][c]

[1]*Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay*

[2]*Pento.ai, Montevideo, Uruguay*

[3]*Centro Interdisciplinario en Ciencia de Datos y Aprendizaje Automático (CICADA), Universidad de la República, Uruguay*

{*federico.bello, gonzalo.chiarlone,mfiori,gastong,flarroca*}*@fing.edu.uy*

Keywords: Multivariate Time Series, Graph Neural Networks, Evaluation Metrics, Score-based Anomaly Detection, Methodological Assessment

Abstract: There is growing interest in applying graph-based methods to Time Series Anomaly Detection (TSAD), particularly Graph Neural Networks (GNNs), as they naturally model dependencies among multivariate signals. GNNs are typically used as backbones in score-based TSAD pipelines, where anomalies are identified through reconstruction or prediction errors followed by thresholding. However, and despite promising results, the field still lacks standardized frameworks for evaluation and suffers from persistent issues with metric design and interpretation. We thus present an open-source framework for TSAD using GNNs, designed to support reproducible experimentation across datasets, graph structures, and evaluation strategies. Built with flexibility and extensibility in mind, the framework facilitates systematic comparisons between TSAD models and enables in-depth analysis of performance and interpretability. Using this tool, we evaluate several GNN-based architectures alongside baseline models across two real-world datasets with contrasting structural characteristics. Our results show that GNNs not only improve detection performance but also offer significant gains in interpretability, an especially valuable feature for practical diagnosis. We also find that attention-based GNNs offer robustness when graph structure is uncertain or inferred. In addition, we reflect on common evaluation practices in TSAD, showing how certain metrics and thresholding strategies can obscure meaningful comparisons. Overall, this work contributes both practical tools and critical insights to advance the development and evaluation of graph-based TSAD systems.

1 INTRODUCTION

Anomaly detection plays a central role in domains such as fraud detection (Hilal et al., 2022), cybersecurity (Siddiqui et al., 2019), industrial monitoring (Nizam et al., 2022), and medical diagnostics (Spence et al., 2001). Within this broad area, Time Series Anomaly Detection (TSAD) focuses on identifying unexpected behaviors in temporally ordered data (Shaukat et al., 2021). In recent years, and driven by its success in other domains, Deep Learning (DL) has been increasingly applied to TSAD (Zamanzadeh Darban et al., 2024).

The typical pipeline for anomaly detection using deep learning consists of two key components: a backbone model and a scoring module (Jin et al., 2024). The backbone is trained under the assumption that most data is normal, and the scoring module flags deviations via reconstruction or prediction errors, serving as proxies for identifying unexpected patterns. However, standard DL models often treat multivariate time series as sequences of independent feature vectors, neglecting structural dependencies that may be essential for accurate and interpretable detection.

Graph Neural Networks (GNNs), designed to operate on graph-structured data, have shown promise for modeling such dependencies. By representing time series as graphs, GNNs enable the joint modeling of temporal dynamics and inter-variable de-

[a] https://orcid.org/0000-0002-3732-1778
[b] https://orcid.org/0009-0002-6652-7713
[c] https://orcid.org/0000-0001-7893-2201

pendencies through message passing, effectively capturing complex relational structures (Chen et al., 2022b; Deng and Hooi, 2021). This ability has spurred a growing interest in graph-based TSAD (Jin et al., 2024), where GNNs act as backbones for reconstruction- or prediction-based anomaly scoring. Yet, despite promising performance, the field remains fragmented: implementations are rarely comparable, evaluation practices vary widely, and metric design often leads to inconsistent or misleading conclusions. As a result, progress is difficult to quantify and reproduce.

To address these issues, we introduce a unified, modular, and open-source framework for graph-based TSAD.[1] Built in PyTorch with reproducibility and extensibility in mind, the framework provides standardized procedures for data handling, model configuration, and evaluation. It natively supports both graph-based and non-graph-based approaches, enabling fair comparisons across modeling paradigms. Crucially, it integrates a diverse set of evaluation metrics, from classical point-wise precision and recall to range-based (Lee et al., 2018; Tatbul et al., 2018) and threshold-agnostic measures such as the Volume Under Surface (VUS) (Paparrizos et al., 2022), offering a consistent environment for methodological analysis.

Using our framework, we conduct a systematic comparative study of representative GNN-based methods and baselines across datasets with contrasting structural characteristics. The results reveal how graph topology, thresholding strategy, and metric design interact to influence performance and interpretability. In particular, we find that attention-based GNNs offer robustness to uncertainty in graph structure while improving interpretability by localizing anomalies to specific nodes. Conversely, we show that common evaluation practices, especially those relying solely on point-wise or threshold-dependent metrics, can obscure genuine model differences.

Beyond empirical benchmarking, this work contributes methodological insights into how graph-based representations and evaluation metrics shape the behavior of TSAD systems. The proposed framework establishes a reproducible foundation for future research in pattern recognition of time series over graphs, facilitating the development of more reliable and interpretable anomaly detection methods.

The rest of this paper is structured as follows. Section 2 formalizes the problem of time series anomaly detection and presents the models and datasets considered in this work. In Sec. 3 we discuss the main methodological challenges in TSAD evaluation, reviewing the limitations of conventional point-wise metrics (e.g., precision and recall), but also their range-based extensions which attempt to account for the temporal extent of anomalies. This section also introduces the proposed framework and its design principles for reproducible experimentation. Equipped with our framework, Sec. 4 presents and discusses the benchmark results obtained with different models, metrics, and graph topologies. Finally, Sec. 5 concludes the article.

[1] The source code and configuration files for our framework are available at `https://github.com/GraGODs/GraGOD`.

2 Problem Statement, Methods and Datasets

Classic TSAD methods had been classified into taxonomies by several authors (Zamanzadeh Darban et al., 2024; Boniol et al., 2024; Blázquez-García et al., 2021). At the coarsest level there are some common groups (eventually overlapping) like Statistical-, Clustering-, Distance-, or Density-based, as well as Forecasting- or Reconstruction-based techniques, which are the focus of this work. Forecasting-based detection builds a model (statistical or machine-learning-based) to predict the next point in the time series. Anomaly scores are then obtained from the residual of the predicted and the real value, and points whose errors exceed a threshold are flagged. Reconstruction-based detection trains an autoencoder, PCA, or matrix-factorization model to compress and then reconstruct windows of observations. If a window cannot be accurately reconstructed (i.e., its reconstruction error is high), the corresponding region is deemed anomalous.

Numerous methods leveraging GNNs for TSAD have been proposed in recent years. For instance, more than thirty such works are discussed in the comprehensive review by (Jin et al., 2024), to which we refer the reader for further details. Across these GNN-based approaches for multivariate TSAD, researchers have explored a diverse set of modeling tools to capture spatiotemporal dependencies and distinguish normal from anomalous behavior. Some methods (Deng and Hooi, 2021; Zhao et al., 2020) learn explicit dependency graphs among variables using attention mechanisms to perform predictive or reconstructive modeling, thereby offering interpretable relations between sensors. Others (Dai and Chen, 2022; Chen et al., 2022a) adopt probabilistic formulations, leveraging normalizing flows for likelihood-based anomaly scoring or combining variational inference with graph convolution and recurrent units to model uncertainty and temporal dynamics. Another

family (Zhang et al., 2022; Han and Woo, 2022) focuses on relational or sparse graph learning, embedding graph-structure discovery directly within autoencoder or forecasting architectures to capture hidden dependencies. We now briefly present how the TSAD problem is formulated in this context, and describe the two state-of-the-art methods of the first family, included in this framework.

Let $\mathcal{X}$ be a set of $N \in \mathbb{N}^*$ distinct time series. We will denote a given time series as $\mathbf{X}_i = [x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(T)}]$, where $x_i^{(t)} \in \mathbb{R}$, and T is the length of the time series. Depending on the application or the labeling of the data, the goal is to detect either an anomaly at a specific time series $i \in [1 \ldots N]$, or an anomaly at a global level, meaning that the system presents an anomaly at a certain time.

The time series may inherently be connected through an underlying graph structure, which may be explicit, as is common in scenarios like sensor networks, or implicit, like causal dependencies in financial markets. In the latter case, a key assumption is that there exists some kind of correlation or dependencies between some of the time series, and therefore a graph can (must) be inferred. In this work we use datasets from both scenarios, presented below. Therefore, given the multivariate set of N time series, we will consider a graph G with N nodes, each of which correspond to a certain time series. The structure of the graph (i.e., the edges and their weights) plays a fundamental role. As mentioned, this structure may come beforehand from the problem itself, like an industrial pipeline with sensors, or may be abstract and learned from the data.

The experimental framework compares multiple GNN-based models, which produce their outputs by operating on the graph through message passing, alongside a structure-agnostic model serving as a benchmark. Each model is trained uniformly, functioning either as a forecaster or reconstructor. The models receive as an input a window of datapoints of size w, defined as:

$$\mathbf{X}^{(t)} = [\mathbf{x}^{(t-w+1)}, \mathbf{x}^{(t-w+2)}, \ldots, \mathbf{x}^{(t)}] \in \mathbb{R}^{N \times w}, \quad (1)$$

where each $\mathbf{x}^{(\tau)}$ is composed of the τ-th datapoint of each time series, i.e., $\mathbf{x}^{(\tau)} = (x_1^{(\tau)}, x_2^{(\tau)}, \ldots, x_N^{(\tau)})$, and produce either an estimate $\hat{\mathbf{x}}^{(t+1)}$ (forecaster) or $\hat{\mathbf{X}}^{(t)}$ (reconstructor).

The approaches based on GNNs compute these outputs by combining the time series using different architectures and supporting graphs. As previously mentioned, the graph can either be learned from the data (e.g., through correlations) or provided by the user. For example, assume we are using a forecasting-based method and we have an actual network G connecting the nodes, which we will represent through the adjacency matrix $\mathbf{A}$ (which may include weights). Then, the trained GNN-based forecaster is a function $\Phi(\mathbf{A}, \theta, \mathbf{X}^{(t)}) = \hat{\mathbf{x}}^{(t+1)}$, where each node has an associated w-dimensional signal as the input, and a 1-dimensional signal as the output. Note that $\mathbf{A}$ is fixed throughout all values of t even if the architecture uses attention mechanisms. Furthermore, if we estimate/infer the graph (and thus the adjacency matrix), we perform this estimation once, meaning that $\hat{\mathbf{A}}$ is also fixed for all values of t (see the discussion in Sec. 4.2).

Given a forecasting- or reconstruction-based method, anomaly scores computed as prediction/reconstruction errors are used to flag an anomaly when it is above a certain threshold. This anomaly may be at the node level (i.e., large errors in predicting/reconstructing an individual time series) or graph level (i.e., considering the error in all time series).

Methods. Currently, our experimental framework incorporates four distinct models. Firstly, a structure-agnostic Gated Recurrent Unit (GRU) and a custom-designed Graph Convolutional Network (GCN), both serving as baselines for comparative analysis. The GCN operates on a fixed given graph structure, and the anomaly scores are computed as forecasting errors.

We have also included two state-of-the-art GNN-based models, which we now briefly describe. The *Graph Deviation Network* (GDN) (Deng and Hooi, 2021) is a deep learning-based approach that learns the structure of dependencies between variables, and using this graph and an attention mechanism produces a prediction of the next value. The *Multivariate Time-series Anomaly Detection via Graph Attention Network* (MTAD-GAT) (Zhao et al., 2020), similar to GDN, also uses a GNN approach. The key of this model is using two different GATs, *feature oriented GAT* and *time oriented GAT*, to map the relationships both between the features and the temporal dependencies. The training involves both a reconstruction and a forecasting model at the same time. The anomaly score is then computed as a combination of both the forecast and reconstruction errors.

Datasets. To evaluate these methods we have chosen two representative datasets: the TELCO and SWaT datasets. The TELCO dataset (González et al., 2024), consists of twelve distinct time series. Each one represent common metrics tracked by a mobile internet service provider (normalized and anonymized), such as the quantity and value of prepaid data transfer fees, the number and cost of calls, the volume of data traffic, and additional related data. The dataset spans seven months, divided into three months for training,

one month for validation, and three months for testing. The most noticeable aspect is the significant imbalance between anomaly and normal data within the dataset, a common characteristic in anomaly detection datasets. A key element is the absence of an explicit graph structure. While the time series data may exhibit correlations, there is no physical structure or defined relationship connecting the series.

The SWaT (Secure Water Treatment) (Goh et al., 2017) dataset is a widely used benchmark for evaluating anomaly detection methods. It consists of time series data collected from a scaled-down six-stage water treatment plant that replicates real-world industrial control systems. The dataset includes 51 physical and network-related features and spans eleven days, where the initial seven days record normal system operations, and the subsequent four days contain both normal and attack scenarios. The attacks were intentionally introduced and encompass both cyber and physical threats to the system. Notably, the SWaT dataset exhibits an inherent relational structure, as sensors within the same treatment stage often measure correlated physical properties like flow rate, pressure, or water level. An undirected graph is constructed where an edge is added between two nodes if the sensors measure similar properties or are in the same stage of the process.

Naturally, both TELCO and SWaT have limitations, including mislabeled anomalies, distribution shifts, and run-to-failure bias, that affect model evaluation. Despite these challenges, they can serve as useful benchmarks when paired with qualitative analysis. Effective preprocessing, such as removing redundancies, fixing labels, and handling distribution shifts, is essential for reliable and fair model comparisons.

3 Challenges in current TSAD methodologies

Point-wise Metrics. Despite the growing interest in TSAD, the evaluation of model performance remains a challenging and often overlooked aspect. Many existing works still rely on Precision, Recall (sensitivity) and F1-score, based on the classification of each time point as either normal or anomalous. However, these point-wise metrics present significant limitations, as they fail to capture the sequential nature and typical range-based structure of anomalies in time series (Tatbul et al., 2018).

In practice, it is often more important to detect as many distinct anomaly ranges as possible, even if their exact boundaries are missed, since identifying the occurrence of each anomaly is typically more valuable than precisely locating every anomalous point. For instance, consider the example in Fig. 1, based on the SWaT dataset which presents a very similar pattern with a single long anomaly and several short ones. In this example, only the long anomaly is correctly detected. Despite this, point-wise metrics report a Recall of 0.8 and a Precision of 1.0, suggesting high performance. In reality, however, the model misses the majority of the anomaly ranges in the dataset, highlighting a major shortcoming of these metrics.

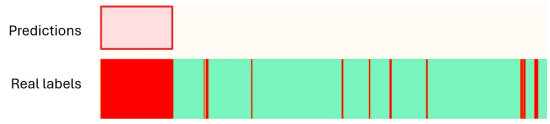

Figure 1: Example of point-wise evaluation limitations. Although only one long anomaly is detected, point-wise metrics report a high Recall (0.8) and perfect Precision (1.0). This gives the false impression of good performance, despite the fact that most anomaly ranges in the dataset remain undetected.

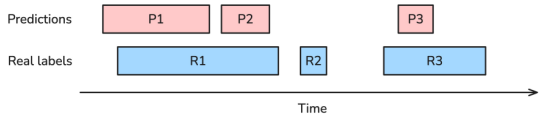

Figure 2: Range-based recall example. R1 and R3 get a high existence reward, R2 none. R1 obtains a high size score, R3 a low one, and R2 none. Cardinality is high for R3 but lower for R1, since the anomaly is detected as two separate segments instead of one. No position reward is considered here.

Range-based metrics. To address these limitations, range-based variants of Precision, Recall, and F1-score (denoted P_T, R_T and $F1_T$ respectively in the sequel) have been proposed (Tatbul et al., 2018). These metrics account for the temporal extent of anomalies by rewarding partial overlap, penalizing fragmentation, and weighting detections by positional relevance, offering a more faithful evaluation than point-wise measures.

For example, range-based Recall R_T evaluates how effectively a detector identifies true anomalous intervals by checking whether an anomaly is detected at all, even if partially (existence reward); how much of its duration is correctly identified (size); which parts are detected, e.g., if early detection is more critical (position); and whether it is reported as a single continuous range or split into fragments (cardinality). An illustrative example is shown in Fig. 2.

These metrics require careful configuration, as their (several) parameters strongly influence results, and poor choices can lead to misleading conclusions. For example, in datasets with long anomalies, neglecting the cardinality component may allow mul-

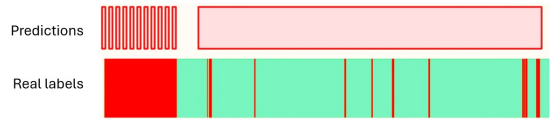

Figure 3: Example of how range-based metrics' configuration can misrepresent performance. The model produces an overall poor prediction: the long anomaly is detected through multiple fragmented predictions, and several extended false positives occur across the timeline. However, under certain range-based metric configurations, such as existence-only recall and precision without cardinality penalty, the evaluation yields a high performance, masking the model's true shortcomings.

tiple overlapping predictions on the same anomaly to artificially boost precision.

Figure 3 illustrates this risk. Here, a long anomaly is detected through several fragmented predictions, with an additional long (and mostly false) detection across the rest of the timeline. Yet, if range-based Recall is configured to reward only the existence of overlap, the score reaches 1.0, since every anomaly is at least partially detected. Likewise, neglecting the cardinality penalty in range-based Precision allows the multiple predictions within the long anomaly to counterbalance the extended false positive, producing a value close to 1.0 despite poor detection quality.

Threshold-agnostic Metrics. To avoid the dependence on a specific value of the threshold, we will use the recently proposed Volume Under Surface (VUS), in its VUS-ROC and VUS-PR variants (Paparrizos et al., 2022). This metric generalizes the traditional AUC concept from binary classification by integrating model performance over multiple buffer sizes around the annotated anomaly ranges. This allows for a continuous assessment of robustness to label imprecision and misalignment. VUS computes the volume under the surface generated by simultaneously varying both the decision threshold and the buffer parameter, eliminating dependence on specific hyperparameters or fixed thresholds. As a result, it provides a more robust and comprehensive evaluation for anomaly detection models.

Lack of framework. One of the most persistent challenges in TSAD research is the absence of a unified, standardized framework for systematically comparing detection methods. Existing implementations are typically tied to specific models, datasets, experimental setups, and metric choices—some of which, as discussed earlier, can produce misleading conclusions. This fragmentation limits reproducibility, constrains the scope of comparative studies, and makes it difficult to assess the real-world applicability of proposed models.

To address this gap, we introduce GraGOD, a modular and extensible open-source framework for the evaluation and comparison of machine learning and deep learning-based TSAD models. Unlike existing solutions (DHI, 2025) that offer limited flexibility, GraGOD is designed as a collaborative, research-oriented framework where new models, datasets, and metrics can be seamlessly integrated. Its architecture natively supports both graph-based and non-graph-based methods, enabling fair and transparent comparison across different paradigms.

GraGOD provides a comprehensive experimental management system for TSAD, with a focus on GNN research. It supports end-to-end experimentation, including data preprocessing, model training, prediction, and hyperparameter tuning. The framework's command-line interface allows users to orchestrate these processes, ensuring reproducibility and control over experiment configuration. GraGOD also integrates automated metric computation of all the metrics mentioned in this work, and provides modules for visualizing anomalies, helping researchers interpret model behaviors beyond numerical scores.

From a development perspective, GraGOD enforces a consistent project structure for code organization, dataset management, and result logging. It supports iterative experimentation and scalable execution, making it suitable for large-scale data analysis and computationally intensive model tuning. The framework's design encourages community contributions by simplifying the addition of new datasets, models, and evaluation metrics through a well-documented API. Ultimately, GraGOD establishes a reproducible and extensible foundation for TSAD research, accelerating methodological progress and promoting transparent, comparable experimentation.

4 Benchmark

This section presents the experimental results, focusing on the impact of graph topology, threshold selection strategies, model interpretability, and the limitations of standard training paradigms. All the models were trained for a maximum of 200 epochs with early stopping, using an initial learning rate of 10^{-3}, with a *reduce on plateau* scheduler with a reduction factor of 0.5. All gradients were clipped to a value of 1.0 to avoid exploding gradients (Pascanu et al., 2012). Thresholds (when used) were chosen to maximize the F1 score on the validation dataset. The configuration of VUS metrics are left as suggested in the original paper (Paparrizos et al., 2022). The hyperparameter optimization was done independently for each model-dataset pair, using TPE sampler algorithm (Parizy et al., 2023) for 100 trials and utilizing

Table 1: Test metrics for the TELCO and SWaT datasets using fully connected graphs.

Dataset	Model	P	R	$F1$	P_T	R_T	$F1_T$	*VUS-ROC*	*VUS-PR*
SWaT	GCN	0.80	**0.79**	0.80	0.06	**0.33**	0.10	0.72	0.55
	GDN	**1.00**	0.75	0.85	**1.00**	0.07	**0.13**	0.85	0.73
	MTAD-GAT	0.00	0.00	0.00	0.00	0.00	0.00	**0.88**	0.74
	GRU	0.98	0.76	**0.86**	0.08	0.24	0.12	0.86	**0.77**
TELCO	GCN	0.15	0.10	0.08	0.09	0.29	0.11	**0.64**	0.05
	GDN	0.32	**0.18**	0.11	0.30	**0.48**	0.25	0.62	0.08
	MTAD-GAT	**0.39**	0.16	0.10	0.34	0.44	0.25	0.61	0.07
	GRU	**0.39**	0.14	**0.12**	**0.35**	**0.48**	**0.30**	0.58	**0.09**

the VUS-ROC as the objective function. Full training details can be consulted on the source code.

4.1 Initial Benchmark Results

Table 1 reports baseline results for the four anomaly detection models on both datasets. In this first experiment, all graph based models used a fully connected graph topology. A first observation is that the GDN, MTAD-GAT, and GRU models consistently achieve similar VUS scores, suggesting their predictive abilities are closely matched. It is interesting to note that VUS metrics are much lower for the TELCO dataset than for SWaT. This is indicative of poor separability between normal and anomalous scores. Since a random classifier yields a VUS-ROC of 0.5, the low VUS-ROC indicates that the models cannot easily distinguish anomalies.

Furthermore, it is important to highlight the clear mismatch between VUS and threshold-dependent metrics: high performance in the former does not necessarily translate into strong performance in the latter. As we discussed before, VUS aggregates performance over all possible thresholds, so it can remain high even if no single threshold yields satisfactory results. An extreme example is MTAD-GAT on SWaT, which achieves highly competitive VUS values yet produces no correct predictions when thresholded. This suggests a threshold selection issue rather than a fundamental model failure. Since the threshold was chosen to maximize $F1$ on the validation set, the problem arises from a shift in the score distribution between validation and test data. This is further confirmed when using more sophisticated selection methods, such as Otsu's algorithm (Yoon et al., 2022), which yields an excellent $F1 = 0.8$ but a disappointing $F1_T = 0.07$. In contrast, a dynamic threshold based on the rolling mean and standard deviation of recent scores produces $F1 = 0.41$ and $F1_T = 0.25$, which, although modest in absolute terms, are notably more consistent with each other than the results obtained with other methods. These findings underscore the critical role of both metric choice and robust thresholding strategies in TSAD evaluation.

The score distributions in Fig. 4 further illustrate these challenges. Each histogram shows the normal

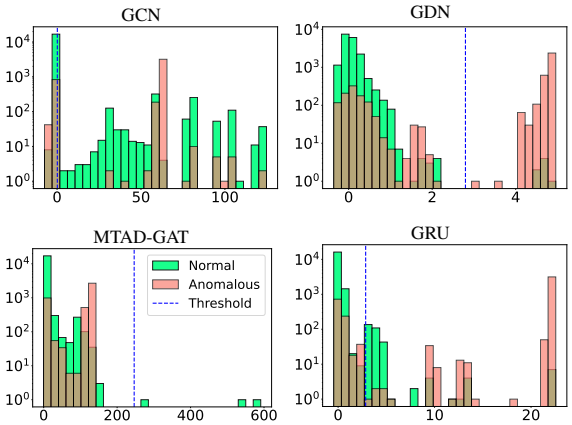

Figure 4: Anomaly score distributions on the SWaT test set for all models (in logarithmic scale). GRU and GDN exhibit better separation between normal (green bars) and anomalous (red) scores, while GCN and MTAD-GAT exhibit significant overlap, complicating threshold selection.

(green) and anomalous (red) scores in the test set for all four methods. GRU and GDN produce relatively well-separated distributions, which helps explain their stronger and more consistent results. In contrast, GCN and MTAD-GAT exhibit substantial overlap between normal and anomalous scores, making reliable threshold selection considerably harder. For MTAD-GAT in particular, the chosen threshold (dashed vertical line) is clearly suboptimal for the test data; an outcome of the distribution shift between validation and test sets. When analyzing the same plot in the TELCO dataset, we observe that the score histograms are not well separated and lack a bimodal structure, which aligns with the lower VUS results reported earlier.

These experiments illustrate a downside of deriving anomaly scores from reconstruction or prediction losses, used as proxies for detecting abnormal behavior, particularly in terms of the threshold selection. While some evaluation metrics such as VUS are threshold-agnostic, real-world applications still require binary decisions, making threshold selection a critical step. We just shown that poor detection performance often stems not from the thresholding method itself (which may even be adaptive), but from the non-discriminative nature of the score distributions produced by certain models. These findings highlight the limitations of proxy-based scoring and point toward the need for more task-aligned objectives, such as learning inherently discriminative representations through, for instance, contrastive learning.

Table 2: VUS metrics (ROC and PR) on test set for different graph topologies in the SWaT and TELCO datasets using the GDN and GCN models. The highest value is shown in **bold** and the second highest is underlined for each model.

Dataset	Model	Graph Topology	*VUS-ROC*	*VUS-PR*
SWaT	GCN	Fully Connected	0.72	0.55
		System Topology	0.79	0.53
		MB	**0.87**	**0.76**
		Random Graph	0.82	0.63
	GDN	Fully Connected	0.82	0.70
		GDN Graph	**0.85**	0.73
		System Topology	**0.85**	**0.75**
		MB	0.83	0.71
		Random Graph	0.83	0.70
TELCO	GCN	Fully Connected	**0.64**	**0.05**
		MB	0.62	0.04
		Random Graph	**0.64**	**0.05**
	GDN	Fully Connected	0.60	0.08
		GDN Graph	0.65	0.08
		MB	0.64	0.05
		Random Graph	**0.67**	**0.09**

4.2 Impact of Graph Topology

We now assess whether the incorporation of a graph structure improves performance by comparing the GCN and GDN models on various topologies: a fully connected graph (as before), the predefined or learned graph (SWaT and GDN only respectively), and finally a statistically inferred graph using the popular Meinshausen-Bühlmann (MB) method (Meinshausen and Bühlmann, 2006).

In the SWaT dataset (see the upper portion of Table 2), which features an underlying physical structure, employing an informative graph topology significantly improves performance, particularly in the GCN case. Note that in the GDN case, although the system topology obtains the best overall results, its attention mechanism makes it robust to the topology's choice. Furthermore, and quiet interestingly, GCN's best results are obtained when using the MB graph and not the system topology. The relationship between variables is thus better captured by the MB method.

On the other hand, in the TELCO dataset no consistent improvement was observed when using different topologies, the best performance being achieved by a random graph. This indicates that using better graph inference methods could lead to improved results, but it is not clear when the dataset does not have an explicit graph structure.

4.3 Metric-Loss correlation

As we discussed before, in many TSAD models, training is performed using regression objectives

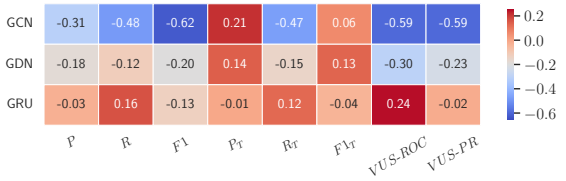

Figure 5: Correlation between the different metrics and the validation loss for the different models.

(e.g., forecasting or reconstruction loss), while evaluation relies on classification metrics. This raises a fundamental question: does minimizing regression loss improve anomaly detection performance?

To explore this, we analyzed the correlation between validation loss, calculated over normal data, and evaluation metrics, computed over the full validation set, including anomalies, across 200 trials from hyperparameter tuning of GCN, GDN, and GRU models on the SWaT dataset. Pearson correlation was used to measure linear relationships, with an ideal scenario corresponding to strong negative correlation (i.e., lower loss leads to better metric scores).

Figure 5 shows the results. The GCN model, which performs worst, exhibits the strongest negative correlation between loss and VUS, suggesting that better forecasting results in an improved anomaly detection. In contrast, the GDN and GRU models achieve superior metric scores but show weak or even positive correlation, implying that a better regression fit does not translate to a better detection performance.

These findings suggest that optimizing purely for regression loss may be suboptimal. Alternative approaches, such as contrastive learning (Liu et al., 2022; Darban et al., 2025), which leverage anomaly labels during training to structure the feature space more effectively, could offer a more aligned and robust solution for anomaly detection tasks.

4.4 Interpretability analysis

Beyond accurate detection, anomaly detection models should help identify where anomalies originate. Graph-based models, especially GDN with attention mechanisms, provide a natural framework for this by modeling sensor dependencies and highlighting influential nodes.

To assess interpretability, we analyze each model's ability to attribute detected anomalies to specific sensors in the SWaT dataset, focusing on a known event affecting sensor FIT401. We compare GDN using both a learned and predefined SWaT topologies, to a the GRU baseline. For GDN, we further examine the distribution of attention weights to identify which sensors most strongly influence the

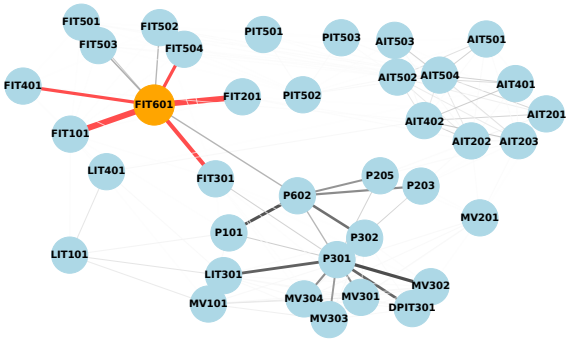

Figure 6: Visualization of attention during the anomaly in FIT401 using the SWaT topology. The node with the highest anomaly score is shown in orange; red edges indicate the five neighbors with the highest attention weights. Attention concentrates on physically connected nodes, improving interpretability and mapping directly to system flow: an anomaly in a FIT sensor.

anomaly score.

All models consistently rank the true anomalous sensor among the top sensors during the anomaly period. This indicates that, while score-based methods can suggest likely affected sensors, interpretability still requires further analysis.

In particular, attention visualization for GDN with the SWaT topology reveals more coherent and physically meaningful patterns. Figure 6 shows attention distributions during the anomaly. The node with the highest anomaly score corresponds to FIT601, while the actual anomaly occurs in FIT401. However, the strongest attention edges are concentrated among FIT sensors (FIT401, FIT101, FIT201), all of which measure related physical quantities. This indicates that the model correctly focuses on a group of related sensors, improving interpretability and helping to identify a consistent set of potentially anomalous sensors. When other graph structures are used, attention becomes dispersed across unrelated nodes, reducing interpretability.

Using a graph topology not only improves attention distributions, but also stabilizes prediction scores, as exemplified in Fig. 7. Here, GDN's graph-based approach keeps forecasts stable and restricts anomaly effects to the affected sensor (PIT501 in stage 5 of the process in this example), making fault localization straightforward. In contrast, GRU forecasts are less stable; an anomaly in stage 5 also deteriorates predictions for stage 2 (P203), making it difficult to pinpoint the true source of the anomaly.

Thus, predefined topologies enhance interpretability by aligning attention with real system structure and stabilizing the predictions of the considered models.

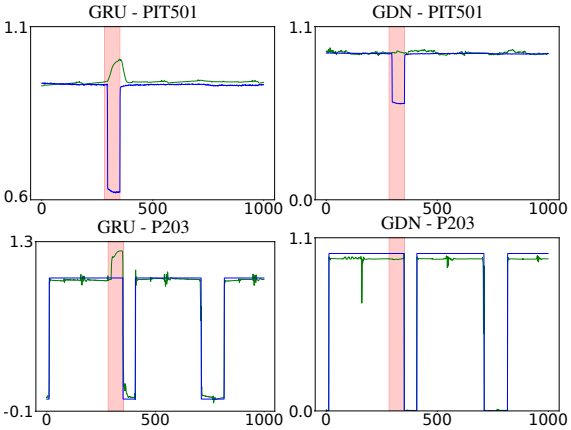

Figure 7: Forecast comparison between GDN with graph topology (right) and GRU (left) on the SWaT dataset. The blue line represents the true values of the time series, the green line shows the forecasted values, and the red shaded regions indicate anomaly labels. GDN's use of a meaningful system topology results in stable forecasts and clear, localized anomaly detection. In contrast, GRU lacks this structure, leading to unstable forecasts and reduced interpretability, as anomalies can affect all sensors.

5 Conclusions

In this work, we introduced a modular and open-source framework for evaluating graph-based models in TSAD. This framework enables reproducible experimentation across datasets, architectures, graph topologies, and evaluation metrics. Using it, we conducted a comparative study of several GNN-based methods and baselines across two real-world datasets with differing structural characteristics.

Our findings show that GNNs can provide competitive, and in some cases superior, performance in TSAD tasks, particularly when there is an underlying and explicit graph. More importantly, they offer improved interpretability by localizing anomalies to specific nodes in the input graph. We also found that attention-based GNNs are more robust to uncertainty in graph construction, making them attractive for use in semi-structured or anonymized datasets. Alongside model evaluation, we critically examined the limitations of commonly used performance metrics and scoring strategies. In particular, we highlighted how score distributions and threshold sensitivity can undermine the reliability of evaluation.

Looking ahead, our findings suggest that moving beyond proxy-based scoring (e.g., reconstruction error) could further improve TSAD systems. In this context, contrastive learning offers a promising direction for producing more discriminative anomaly scores directly aligned with the detection task (Liu et al., 2022; Darban et al., 2025), which we plan to explore and integrate to our framework in future work.

REFERENCES

Blázquez-García, A., Conde, A., Mori, U., and Lozano, J. A. (2021). A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3).

Boniol, P., Liu, Q., Huang, M., Palpanas, T., and Paparrizos, J. (2024). Dive into time-series anomaly detection: A decade review. *arXiv preprint arXiv:2412.20512*.

Chen, W., Tian, L., Chen, B., Dai, L., Duan, Z., and Zhou, M. (2022a). Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 3621–3633. PMLR.

Chen, Z., Chen, D., Zhang, X., Yuan, Z., and Cheng, X. (2022b). Learning graph structures with transformer for multivariate time-series anomaly detection in iot. *IEEE Internet of Things Journal*.

Dai, E. and Chen, J. (2022). Graph-augmented normalizing flows for anomaly detection of multiple time series. In *International Conference on Learning Representations*.

Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C. C., and Salehi, M. (2025). Carla: Self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognition*.

Deng, A. and Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. In *AAAI conference on artificial intelligence*.

DHI (2025). tsod: Anomaly detection for time series data. `https://github.com/DHI/tsod`. Accessed: 2025-08-14.

Goh, J., Adepu, S., Junejo, K. N., and Mathur, A. (2017). A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security*, pages 88–99.

González, G. G., Tagliafico, S. M., Fernández, A., Sena, G. G., Acuña, J., and Casas, P. (2024). One model to find them all deep learning for multivariate time-series anomaly detection in mobile network data. *IEEE Transactions on Network and Service Management*.

Han, S. and Woo, S. S. (2022). Learning sparse latent graph representations for anomaly detection in multivariate time series. In *Proceedings of the 28th ACM SIGKDD Conference on knowledge discovery and data mining*, pages 2977–2986.

Hilal, W., Gadsden, S. A., and Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications*, 193:116429.

Jin, M., Koh, H. Y., Wen, Q., Zambon, D., Alippi, C., Webb, G. I., King, I., and Pan, S. (2024). A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lee, T. J., Gottschlich, J., Tatbul, N., Metcalf, E., and Zdonik, S. (2018). Precision and recall for range-based anomaly detection. In *Proceedings of the SysML Conference 2018*.

Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., and Karypis, G. (2022). Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3).

Nizam, H., Zafar, S., Lv, Z., Wang, F., and Hu, X. (2022). Real-time deep anomaly detection framework for multivariate time-series data in industrial iot. *IEEE Sensors Journal*, 22(23):22836–22849.

Paparrizos, J., Boniol, P., Palpanas, T., Tsay, R. S., Elmore, A., and Franklin, M. J. (2022). Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *Proceedings of the VLDB Endowment*.

Parizy, M., Kakuko, N., and Togawa, N. (2023). Fast hyperparameter tuning for ising machines. *2023 IEEE International Conference on Consumer Electronics*, pages 1–6.

Pascanu, R., Mikolov, T., and Bengio, Y. (2012). On the difficulty of training recurrent neural networks.

Shaukat, K., Alam, T. M., Luo, S., Shabbir, S., Hameed, I. A., Li, J., Abbas, S. K., and Javed, U. (2021). A review of time-series anomaly detection techniques: A step to future perspectives. In Arai, K., editor, *Advances in Information and Communication*, pages 865–877.

Siddiqui, M. A., Stokes, J. W., Seifert, C., Argyle, E., McCann, R., Neil, J., and Carroll, J. (2019). Detecting cyber attacks using anomaly detection with explanations and expert feedback. In *ICASSP 2019*.

Spence, C., Parra, L., and Sajda, P. (2001). Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *MMBIA 2001*.

Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., and Gottschlich, J. (2018). Precision and recall for time series. *Advances in neural information processing systems*, 31.

Yoon, J., Sohn, K., Li, C.-L., Arik, S. O., Lee, C.-Y., and Pfister, T. (2022). Self-supervise, refine, repeat: Improving unsupervised anomaly detection. *Transactions on Machine Learning Research*.

Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., and Salehi, M. (2024). Deep learning for time series anomaly detection: A survey. *ACM Comput. Surv.*, 57(1).

Zhang, W., Zhang, C., and Tsung, F. (2022). Grelen: Multivariate time series anomaly detection from the perspective of graph relational learning. In *IJCAI*, pages 2390–2397.

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 *ICDM 2020*.