C03025: Master of Science (Research)
CRICOS Code: 001121E
31675 Thesis: Computing Science
June 2025

Spatial Temporal Graph Neural Networks based Model for Anomaly Detection

Zhiqin Tang

School of Computer Science
Faculty of Engineering. & IT
University of Technology Sydney
NSW - 2007, Australia

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Zhiqin Tang, declare that this thesis is submitted in fulfilment of the requirements for the award of Master of Science, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220.

SIGNATURE: Production Note: Signature removed prior to publication.

DATE: 10th June, 2025

PLACE: Sydney, Australia

Abstract

The rapid development of the cyber-physical systems has led to the deployment of massive sensor networks that continuously generate large volumes of multivariate time-series data. In industrial environments, where reliability and operational safety are paramount, effective anomaly detection is essential for ensuring system stability and early fault prevention. However, conventional statistical and deep learning approaches often fail to capture the complex spatial and temporal dependencies that exist among heterogeneous sensors, particularly in industrial control systems (ICS) where operating conditions are dynamic and anomalies are rare. The diversity of sensor behaviours, high dimensionality, and the lack of labelled data further exacerbate the difficulty of building accurate and generalisable detection frameworks.

To address these challenges, this research proposes MLAD (Multi-task Learning-based Anomaly Detection), a novel spatial-temporal graph neural network (STGNN) framework that unifies unsupervised sensor clustering, graph structure learning, and multi-task forecasting. The framework begins by learning low-dimensional temporal embeddings of sensor behaviour using UMAP-based dimensionality reduction, followed by DBSCAN clustering to group sensors exhibiting similar temporal dynamics. These clusters are then used to form a cluster-constrained adjacency matrix, ensuring that graph learning is both physically meaningful and computationally efficient. The core architecture integrates Graph Convolutional Networks (GCN) for modelling spatial dependencies among sensors and Temporal Convolutional Networks (TCN) for capturing sequential temporal patterns. Furthermore, MLAD introduces a multi-task learning mechanism that simultaneously performs forecasting and reconstruction tasks. This design allows the model to capture both short-term, abrupt deviations and long-term, gradual drifts, thereby enhancing its robustness and interpretability in detecting diverse anomaly types.

Extensive experiments were conducted on two benchmark ICS datasets, SWaT and WADI, to evaluate the proposed framework. The results demonstrate that MLAD consistently outperforms state-of-the-art baselines in both ROC-AUC and PRC-AUC metrics. The ablation study confirms the importance of clustering-guided graph learning and the contribution of the multi-task design, while the case studies visually reveal the complementary strengths of the forecasting and reconstruction heads: the forecasting component excels at identifying sudden anomalies, whereas the reconstruction component effectively captures persistent, low-amplitude drifts.

In summary, this research presents a unified, interpretable, and generalisable ap-

proach to anomaly detection in multivariate time-series data. By integrating spatial-temporal graph learning with multi-task representation learning, MLAD advances the state of the art in intelligent fault diagnosis and resilient monitoring for large-scale industrial systems, offering new insights into reliable, data-driven industrial analytics.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisors, Dr. Bin Liang, Dr. Shuming Liang and Prof. Zhidong Li for their invaluable guidance, continuous support, and insightful feedback throughout the course of this research. Their expertise and patience were instrumental in the completion of this thesis.

I am also thankful to the University of Technology Sydney (UTS) and the Australian Government Research Training Program for providing the financial support and resources that made this work possible.

My sincere thanks go to my colleagues and collaborators for their stimulating discussions and encouragement.

Lastly, I wish to thank my family and friends for their unwavering support and understanding during this journey.

List of Publications

Related to the Thesis :

1. Li, Kunqi, Zhiqin Tang, Shuming Liang, Zhidong Li, and Bin Liang. 2025. "MLAD: A Multi-Task Learning Framework for Anomaly Detection" Sensors 25, no. 13: 4115. https://doi.org/10.3390/s25134115

TABLE OF CONTENTS

List of Publications **ix**

List of Figures **xiii**

List of Tables **xv**

1 Introduction **1**

1.1 Background 1
1.2 Motivation 2
1.3 Challenges 3
1.4 Contributions 5
1.5 Thesis Structure 6

2 Literature Review **7**

2.1 Anomaly Detection 7
2.2 Sensor Clustering 9
2.3 Graph Neural Networks for Anomaly Detection 12
2.4 Spatial Temporal Graph Neural Networks for Anomaly Detection 14
2.5 Datasets for Multivariate Time-Series Anomaly Detection 16
2.6 Summary 18

3 Sensor Embedding and Clustering **19**

3.1 Data Preprocessing and Feature Embedding 20
3.2 Introduction to UMAP 21
3.3 Clustering Method Selection and DBSCAN Implementation 22
3.4 Alignment Between Clusters and Physical System Structure 24
3.5 UMAP-Based Temporal Embedding 25
3.6 Evaluation and Ablation Analysis 26
3.7 Summary 27

4 Spatial Temporal GNN Modelling **29**
4.1 Overview of the Framework 30
4.2 Clustering Graph Construction 31
4.3 Spatial Dependency Encoding via GCN 33
4.4 Temporal Dependency Encoding via TCN 34
4.5 Prediction Heads and Training Objective 35
4.6 Training Strategy and Optimisation 37
4.7 Summary 37

5 Multi-task Forecasting **39**
5.1 Motivation for Multi-task Forecasting 40
5.2 Architecture Design of Multi-task Heads 41
5.3 Loss Function and Training Objective 43
5.4 Quantitative Evaluation and Anomaly Scoring 44
5.5 Case Studies on Forecasting and Reconstruction 46
5.6 Comparison with Existing Forecasting-based Models 49
5.7 Summary 50

6 Conclusion **53**

Bibliography **57**

List of Figures

Figure **Page**

3.1 Examples of sensor behaviour patterns in the WADI (left) and SWaT (right) datasets. Sensors exhibit diverse dynamics such as oscillation, drift, or sudden transitions, which motivate unsupervised grouping. 21

3.2 Performance comparison of the MLAD model using K-Means and DBSCAN clustering on the SWaT and WADI datasets. DBSCAN consistently outperforms K-Means, demonstrating its robustness in modelling complex sensor behaviour. 24

3.3 Mapping between clustered sensor groups and physical system layout. Left: simplified SWaT layout with sensors grouped around Tank 1, Tank 2, and Pump P2. Right: DBSCAN clusters in UMAP-embedded space. 25

3.4 Ablation study results comparing the full model (MLAD) and its variant without the clustering module. The full version consistently outperforms in both ROC AUC and PRC AUC across SWaT and WADI datasets. 26

4.1 Overall architecture of the MLAD model. The framework consists of a clustering-based graph encoder, a temporal convolutional module, and dual-task output heads. 31

4.2 Example of a cluster-constrained adjacency matrix used in MLAD. Each block corresponds to a fully connected subgraph of behaviourally similar sensors. . 32

4.3 GCN-based spatial feature propagation within sensor clusters. (**a**) Cluster architecture showing three distinct sensor groups with intra-cluster connectivity. (**b**) Two-layer GCN message passing process demonstrating feature aggregation from neighbouring sensors within the same cluster. The mathematical formulation illustrates the layer-wise propagation rule. 34

4.4 Architecture of the Temporal Convolutional Network (TCN) with stacked dilated convolutional layers. The network processes input sequences through multiple hidden layers with exponentially increasing dilation rates ($r = 1, 2, 4$), enabling the capture of multi-scale temporal dependencies while maintaining a compact architecture. Each layer's dilation rate doubles, exponentially expanding the receptive field to capture both short-term and long-range patterns in the sensor data. 35

5.1 Illustration of anomaly detection using forecasting (left) and reconstruction (right). The forecasting-based method captures abrupt spikes and collective anomalies, while the reconstruction-based method effectively detects contextual anomalies over extended periods. This motivates the use of both tasks in MLAD. 41

5.2 Case 1: Case study on a WADI sensor showing a spike anomaly. The forecasting head captures the sudden deviation, while the reconstruction head fails to respond to the abrupt change. The shaded region indicates the anomalous period. 47

5.3 Case 2: A sensor showing a slow drift anomaly. The reconstruction head accumulates greater deviation from the ground truth than the forecasting head, highlighting its effectiveness in capturing persistent changes. The shaded region indicates the anomalous period. 48

List of Tables

Table **Page**

2.1 Statistics of benchmark datasets for multivariate time-series anomaly detection. 18

5.1 Ablation study: AUC performance of different task configurations. Best results in each column are **bolded**. 46

5.2 Average AUC performance (± standard deviation) of five experimental runs on three benchmark datasets. The best, second best, and third best results in each column are marked in **bold**, with and † respectively. 50

CHAPTER 1

INTRODUCTION

1.1 Background

Anomaly detection in multivariate time-series data has emerged as a critical task across a wide range of application domains, including industrial monitoring [36], power systems [25, 57], healthcare [16], and cyber-physical systems [13]. As sensor networks in such systems become increasingly complex and interconnected, the volume and dimensionality of time-series data continue to grow, making it significantly more challenging to detect subtle and heterogeneous abnormal behaviours [17, 51]. In real-world settings, anomalies may correspond to system faults, cyberattacks, equipment malfunctions, or unexpected external interventions [13]. Detecting these anomalies accurately and in a timely manner is vital for maintaining system reliability, safety, and performance.

Traditional statistical or threshold-based methods often fail in high-dimensional environments due to their assumptions of linearity, independence, or static distributions [51]. Meanwhile, machine learning-based approaches offer a more flexible framework by learning patterns from data. Yet, most conventional deep learning models treat the input variables as independent sequences, ignoring the structural relationships among sensors [55]. This limits their effectiveness in modelling inter-dependencies that are often essential in characterising system dynamics.

To overcome this limitation, graph-based models have been proposed to capture correlations among variables [15]. Graph Neural Networks (GNNs), in particular, have

demonstrated remarkable success in modelling structured data [60] and learning spatial dependencies [50]. When combined with temporal modelling modules such as recurrent networks or temporal convolutions, these models commonly referred to as Spatial-Temporal Graph Neural Networks (STGNNs) are well-suited for time-series anomaly detection tasks in sensor networks [18, 52, 62].

The increasing deployment of sensor networks in Industry 4.0 environments further necessitates the development of intelligent monitoring techniques that are scalable, adaptive, and robust [55]. These systems are characterised by the heterogeneity of sensor modalities, dynamic system behaviours, and varying fault frequencies. For instance, in manufacturing plants, a single production line may integrate temperature, vibration, and acoustic sensors with different sampling rates and physical units [40]. In process industries such as water treatment and chemical refining, multistage control loops and feedback mechanisms lead to strong temporal dependencies and non-stationary behaviours [3, 43]. Similarly, in smart power grids, the frequency of transient faults and load fluctuations varies over time, posing challenges for scalable and adaptive fault detection [25, 57]. Therefore, an effective anomaly detection method must not only model temporal dependencies but also exploit spatial correlations and structural modularity of the system.

1.2 Motivation

Although STGNNs have shown promise in MTSAD (Multivariate Time-Series Anomaly Detection) [51], they typically rely on static, predefined graph structures or assume homogeneity across sensors. This simplification is problematic in real-world Industry scenarios, where sensors are deployed across subsystems with distinct physical, functional, or operational characteristics [63]. As a result, a single, unified prediction function often fails to accurately capture the heterogeneous temporal behaviours present in different parts of the system.

Recent developments suggest that explicitly modelling modularity and heterogeneity within sensor networks can significantly improve anomaly detection performance. One promising direction involves identifying groups of behaviourally similar sensors and assigning them to separate modelling or forecasting components. This design allows the model to preserve local correlations while reducing interference from unrelated sensor dynamics, which is particularly important in large-scale industrial systems with complex subsystem interactions [18, 36, 63]. Cluster-aware modelling not only enhances robust-

ness against localised or cluster-specific anomalies but also improves interpretability by aligning detected anomalies with specific functional subsystems [15]. Furthermore, many real-world systems lack prior knowledge of sensor groupings or hierarchical structures, especially when dealing with third-party deployments or legacy instrumentation. In these scenarios, data-driven methods for discovering latent modular structures become essential. In particular, the idea of using unsupervised clustering algorithms such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [47] on learned temporal embeddings has emerged as an effective way to discover such groupings. These cluster structures can be further integrated into graph construction and forecasting architectures to constrain the learning process and improve generalisation.

In addition to modelling spatial-temporal dependencies, another key motivation of this research lies in enhancing anomaly detection through multi-task learning (MTL). In industrial control systems, anomalies may manifest either as sudden transient deviations or as slow contextual drifts, each requiring different temporal sensitivities. Single-objective models tend to overfit to one anomaly pattern and fail to generalise across diverse behaviours [30, 34]. By jointly optimising forecasting and reconstruction objectives, MTL enables the model to capture both short-term predictive inconsistencies and long-term structural deviations [21]. This dual-head design allows the forecasting branch to act as a fine-grained anomaly detector for abrupt changes, while the reconstruction branch provides a smoother estimate that accumulates error when system dynamics drift. Similar hybrid paradigms have been shown to significantly improve robustness and interpretability in multivariate time-series anomaly detection tasks [27, 36]. Therefore, integrating MTL within the proposed spatial-temporal framework offers a unified approach to address the heterogeneous nature of anomalies observed in complex industrial environments.

Motivated by these findings, this study seeks to investigate the integration of sensor clustering and multi-task learning into a forecasting-based STGNN anomaly detection framework. The proposed architecture, MLAD (Multi-task Learning for Anomaly Detection), is designed to address the complexity and variability inherent in real-world sensor networks, offering an interpretable and modular approach to MTSAD.

1.3 Challenges

Despite the remarkable progress in deep learning based anomaly detection, several key challenges remain when applying these techniques to complex industrial control systems

(ICS). These challenges form the motivation and foundation of this research and are systematically addressed by the proposed MLAD framework.

(1) Heterogeneous and dynamic sensor behaviours. Industrial systems consist of multiple subsystems, each equipped with sensors operating under distinct physical processes, sampling rates, and noise characteristics. This heterogeneity makes it difficult for a single model to capture the full range of temporal patterns. Conventional methods often assume homogeneous dynamics and therefore suffer from reduced generalisation when applied to mixed sensor types.

(2) Absence of reliable structural priors for graph learning. Most graph-based models rely on predefined or static adjacency matrices derived from physical layouts or correlation analysis. Such graphs rarely align with the true functional dependencies among sensors, leading to either redundant or spurious connections. Learning a meaningful graph structure directly from data,Äîwhile maintaining interpretability remains a non-trivial task.

(3) Limited labelled data and noisy measurements. Real-world ICS data are predominantly unlabelled, with anomalies being rare and irregular. In addition, sensor readings often contain noise, missing values, or calibration drift, all of which can distort statistical learning. Training robust models in such environments requires unsupervised or self-supervised approaches capable of extracting stable representations from normal behaviour alone.

(4) Diverse temporal signatures of anomalies. Anomalies in industrial systems do not follow a single pattern. Some occur as sudden spikes or short-lived transients, whereas others appear as gradual drifts over extended periods. Single-objective models,Äîbased solely on forecasting or reconstruction struggle to capture this diversity. A unified approach that can respond to both abrupt and slow-changing anomalies is therefore essential.

Addressing these challenges requires a framework that can discover latent sensor structures, learn meaningful spatial-temporal dependencies without supervision, and remain sensitive to multiple anomaly types. The MLAD model proposed in this thesis directly targets these issues through a combination of unsupervised clustering, graph learning, and multi-task optimisation.

1.4 Contributions

To overcome the above challenges, this research presents **MLAD (Multi-task Learning based Anomaly Detection)**, a unified spatial-temporal graph neural network framework for robust and interpretable anomaly detection in multivariate time-series data. The principal contributions of this work are summarised as follows:

1. **Unsupervised sensor behaviour clustering.** A data-driven clustering pipeline combining Uniform Manifold Approximation and Projection (UMAP) with Density-Based Spatial Clustering (DBSCAN) is proposed to extract low-dimensional embeddings of sensor dynamics. This enables automatic discovery of behaviourally coherent sensor groups without prior system knowledge.

2. **Cluster-constrained graph construction.** A novel graph learning strategy is designed, in which sensors within the same cluster form fully connected subgraphs while inter-cluster links are masked. This structure ensures that message passing in the graph neural network is physically meaningful, reduces over-smoothing, and improves computational efficiency.

3. **Spatial-temporal modelling with multi-task learning.** The MLAD architecture integrates Graph Convolutional Networks (GCN) for spatial dependency encoding and Temporal Convolutional Networks (TCN) for sequential pattern learning. Two prediction heads forecasting and reconstruction are jointly optimised through a multi-task objective, enabling the model to capture both short-term and long-term anomaly signatures.

4. **Comprehensive evaluation and interpretability.** Extensive experiments on two benchmark ICS datasets, SWaT and WADI, demonstrate that MLAD consistently outperforms state-of-the-art baselines in ROC-AUC and PRC-AUC metrics. Ablation studies further confirm the effectiveness of clustering-guided graph learning and the complementary strengths of the dual-task design, offering interpretable insights into different anomaly behaviours.

In summary, MLAD provides a cohesive solution that bridges unsupervised structural discovery and multi-task temporal learning. By addressing the heterogeneity of sensor behaviour, the absence of reliable structural priors, and the diversity of anomaly patterns, this work advances the state of the art in data-driven anomaly detection for complex industrial systems.

1.5 Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2: Literature Review.** This chapter surveys existing research on anomaly detection, sensor clustering, GNN-based frameworks, and spatial-temporal modelling techniques, along with a discussion of relevant benchmark datasets.
- **Chapter 3: Sensor Embedding.** This chapter introduces the method for extracting meaningful low-dimensional representations from multivariate time-series data and discusses how these representations are used to derive behaviourally coherent sensor clusters.
- **Chapter 4: Spatial temporal GNN Modelling.** This chapter presents the spatial-temporal modelling pipeline built on a cluster-constrained graph neural network and temporal convolutions for learning dynamic inter-sensor dependencies.
- **Chapter 5: Multi-task Forecasting.** This chapter details the design of a multi-task learning architecture that utilises sensor clusters to build task-specialised forecasting heads and evaluates their impact on anomaly scoring.
- **Chapter 6: Conclusion.** This chapter summarises the key contributions of the thesis and discusses future directions for improving modular and interpretable anomaly detection frameworks.

CHAPTER 2

LITERATURE REVIEW

This chapter surveys the literature on Multivariate Time-Series Anomaly Detection (MTSAD), with a focus on techniques that address the challenges of modelling complex spatial and temporal dependencies in Industrial Control Systems (ICS). The review is organised into five key sections. 1) I begin by surveying foundational concepts and traditional approaches to anomaly detection, setting the stage for understanding their limitations in modern high-dimensional settings. 2) I examine sensor clustering methods, which play a critical role in partitioning high-dimensional sensor streams into behaviourally coherent groups. 3) I explore GNNs (Graph Neural Networks) and their extensions as a means to model structured relationships among sensors. 4) I continue with a discussion of STGNNs (Spatial-Temporal Graph Neural Networks), which unify spatial correlations and temporal dynamics within a graph-based learning framework. 5) I summarise key benchmark datasets used in anomaly detection research to highlight the real-world challenges and evaluation practices that inform model design.

2.1 Anomaly Detection

Anomaly detection, also known as outlier detection, refers to the identification of patterns in data that deviate from expected behaviour. These anomalies often signify important events such as system malfunctions [13], fraudulent activities [4, 9], or cybersecurity threats [28], and their early detection is essential in domains like industrial monitoring

[3, 43], finance [12], and healthcare [16].

Anomalies may manifest in various forms, including individual data points that differ from the norm (point anomalies), values that are only abnormal within certain contexts such as time or location (contextual anomalies), and sequences of otherwise normal data points that collectively form an unusual pattern (collective anomalies) [8, 13]. These distinctions are crucial in time-series settings, where the temporal structure and contextual dependencies significantly influence anomaly characterisation.

Conventional anomaly detection methods predominantly relied on statistical modelling and distance-based heuristics. Statistical techniques, such as z-scores, autoregressive models (ARIMA) [6], or PCA-based reconstruction [48], assumed that normal data followed specific distributions [1], whereas distance-based methods identified anomalies as observations located far from clusters or nearest neighbours [22]. Density-based methods such as LOF (Local Outlier Factor) evaluated the local neighbourhood density to detect deviations [10]. Although these methods demonstrated effectiveness in well-defined, low-dimensional settings, they often fail in high-dimensional or temporally complex scenarios due to their linearity assumptions and sensitivity to parameter tuning [46].

With the rise of machine learning, both supervised and unsupervised anomaly detection methods have become prevalent. Supervised methods like support vector machines (SVMs) [48] and ensemble classifiers require labelled anomalies [8], which are often scarce or costly to obtain in real-world applications [42, 46]. Consequently, unsupervised methods, which infer normal patterns from unlabelled data, have become a practical alternative. Principal Component Analysis (PCA) [48], Isolation Forests [38], and clustering-based algorithms [11] exemplify this approach, each leveraging assumptions about the compactness or structure of normal data distributions.

The advent of deep learning introduced powerful tools for anomaly detection, particularly in high-dimensional and sequential data domains. Recurrent neural networks (RNNs) [18], especially LSTM (Long Short-Term Memory) variants [29], are effective in capturing long-term dependencies in time series, while autoencoders reconstruct input sequences and flag those with high reconstruction error as anomalous [5, 49]. GAN-based (Generative Adversarial Network) methods have also been explored for their ability to model data distributions and detect anomalies by comparing generated and real samples [34, 59]. However, despite their expressive power, deep learning models often suffer from limited interpretability and are sensitive to training data quality [12].

Multivariate time-series anomaly detection poses additional complexities beyond

those encountered in uni-variate cases. High dimensionality increases the difficulty of modelling cross-variable interactions, and the heterogeneity of sensor modalities introduces noise and variability in behaviour. Furthermore, anomalies in such systems are often sparse and imbalanced, which challenges both learning and evaluation [17]. These considerations highlight the need for models that can simultaneously model spatial structures and temporal sequences while maintaining robustness and scalability [46].

Recent work has increasingly focused on incorporating structural priors and relational reasoning into anomaly detection frameworks. Graph-based approaches have emerged as a promising direction, enabling the representation of inter-sensor dependencies through graph topologies [18, 55, 63]. GNNs, in particular, allow message passing across nodes and can be combined with temporal modules to model spatial temporal phenomena. This shift toward structured models marks a critical development in the field, aiming to address the limitations of purely sequential or purely statistical techniques [64].

In summary, traditional anomaly detection methods ranging from statistical and distance-based techniques to advanced deep learning architectures have demonstrated significant progress in identifying abnormal patterns across diverse domains. However, most existing approaches treat multivariate time-series data as a unified feature space, overlooking the modular and hierarchical nature of complex industrial systems [18, 26, 63]. In industrial control scenarios, sensors often operate under distinct physical processes and exhibit heterogeneous behaviours, meaning that anomalies may emerge locally within specific subsystems rather than globally across the entire network [3, 43]. Consequently, learning meaningful inter-sensor relationships and grouping sensors with similar temporal or functional characteristics becomes a critical step toward improving detection accuracy, scalability, and interpretability. This motivates the exploration of sensor clustering techniques, which can capture the latent modular structure of sensor networks and serve as a foundation for adaptive graph-based anomaly detection frameworks, as discussed in the following section.

2.2 Sensor Clustering

Sensor clustering has emerged as a pivotal strategy in multivariate time-series anomaly detection due to the increasingly high dimensionality and structural heterogeneity of real-world sensor networks. In practical deployments such as industrial control systems [3, 43], power grids [57], and environmental monitoring [61], sensors are distributed across various subsystems and geographical zones, each operating under distinct dynamics and

functional objectives. Aggregating all sensors into a single monolithic model often leads to performance degradation, as such models fail to account for inter-sensor variability and localised behaviours. Clustering provides a principled way to group sensors with similar temporal patterns, enabling more targeted and specialised analysis.

One of the most common motivations behind sensor clustering is to improve model interpretability and robustness. Grouping similar sensors allows anomaly detection systems to isolate and analyse faults within specific subsystems, which is particularly beneficial in large-scale deployments where interpretability is essential for diagnosis and remediation. Moreover, clustering enhances generalisation by allowing models to learn separate representations for each sensor group, reducing the risk of overfitting to dominant patterns and mitigating the influence of noise from unrelated sensors.

Clustering methods employed in anomaly detection can be broadly classified into two categories: feature-based clustering and behaviour-based clustering. Feature-based methods group sensors based on static attributes such as location, type, or domain-specific metadata. These approaches are straightforward to implement but may not fully capture the operational dynamics of the sensors. In contrast, behaviour-based clustering focuses on the temporal characteristics of sensor readings, grouping sensors that exhibit similar time-series patterns. Dynamic Time Warping (DTW), autocorrelation-based distances, and statistical similarity metrics are commonly used to compute pairwise distances in such settings [37].

Despite their effectiveness, distance-based clustering methods often suffer from performance degradation in high-dimensional settings, where the curse of dimensionality renders Euclidean or correlation-based distances less meaningful and amplifies the influence of noise [2, 7]. To address the limitations of distance-based clustering in high-dimensional settings, dimensionality reduction techniques have been introduced to extract meaningful features from raw time series. Manifold learning approaches, particularly Uniform Manifold Approximation and Projection (UMAP) [44], have demonstrated significant utility in this regard [50]. UMAP embeds high-dimensional sequences into a lower-dimensional latent space where temporal correlations and local neighbourhood structures are preserved. This transformation facilitates downstream clustering by making the geometry of the data more tractable [44].

Once embeddings are computed, clustering can be applied using algorithms such as k-means or agglomerative hierarchical clustering. K-means clustering [39] partitions sensors into a predefined number of clusters by minimising within-cluster variance, offering a simple and efficient solution. However, it is sensitive to initialisation and assumes

spherical cluster shapes. Hierarchical clustering [45], on the other hand, does not require a predetermined number of clusters and can produce a tree of sensor groupings, from which a suitable partition can be selected based on application-specific criteria.

In systems with poorly understood topology or limited metadata, clustering offers a data-driven alternative to manual subsystem definitions. In such cases, clustering acts as an unsupervised prior, helping to uncover latent structures in sensor networks. This is particularly advantageous in large-scale Internet of Things deployments, where sensors may be added or removed dynamically and labelled data are scarce. The resulting clusters can then be used to inform the design of prediction heads, graph construction, or model interpretability modules [15].

Recent anomaly detection models have integrated clustering more tightly into their learning pipelines. Instead of treating clustering as a preprocessing step, these models adaptively learn sensor groups during training. For example, temporal embeddings can be refined end-to-end by optimising forecasting or reconstruction objectives, and clustering can be performed in the learned latent space. This joint optimisation framework ensures that the clusters are not only structurally coherent but also relevant to the downstream detection tasks [36].

Additionally, several models employ clustering to constrain graph construction in spatial-temporal learning architectures. By allowing graph edges only within clusters, these methods limit message passing to semantically similar nodes, which prevents over-smoothing and improves generalisation [63]. This practice is particularly effective in deep graph models, where unrestricted aggregation across unrelated sensors can dilute important signals.

In practice, sensor clustering has been shown to improve anomaly detection performance across several benchmark datasets [3, 43]. It allows the model to focus on more homogeneous subpopulations of sensors, leading to better representation learning and prediction accuracy. Clustering also supports modular deployment, where updates or retraining can be restricted to specific clusters, reducing the overall system complexity and improving scalability.

As multivariate time-series datasets continue to grow in dimensionality and diversity, the importance of sensor clustering will only increase. Advances in unsupervised learning, representation learning, and scalable clustering algorithms will likely drive the development of more sophisticated clustering-based anomaly detection frameworks. In particular, hybrid approaches that combine clustering with attention mechanisms, graph structures, or contrastive learning offer promising directions for future research [15].

In summary, sensor clustering serves as an essential mechanism for structuring high-dimensional sensor networks, enhancing model granularity, and enabling scalable, interpretable anomaly detection. Its ability to capture functional modularity, reduce interference, and improve learning efficiency makes it a critical component in modern MTSAD systems.

2.3 Graph Neural Networks for Anomaly Detection

GNNs have emerged as powerful tools for learning over structured data, offering a compelling framework for modelling the inter-variable dependencies intrinsic to multivariate time-series anomaly detection tasks [18, 55]. In systems where sensor variables are inherently interconnected either through physical topology, functional relationships, or correlated behaviours, GNNs offer a data-driven means of capturing these spatial and temporal dependencies in a unified manner [64]. This capability has made GNN-based models an increasingly prominent choice in the field of anomaly detection across CPS, finance, healthcare, and industrial automation [52, 63].

The primary advantage of GNNs lies in their ability to perform message passing on graphs, enabling each node (sensor) to aggregate and transform information from its neighbours. This is particularly beneficial for anomaly detection, where context plays a crucial role in identifying irregular patterns. For instance, a temperature reading that might appear anomalous in isolation could be normal when seen in the context of pressure or humidity readings from nearby sensors. GNNs model such contextual relationships through edge-defined neighbourhood aggregation, allowing for more nuanced decision-making [58].

One of the earliest applications of GNNs to anomaly detection is found in the work of Ding et al. [18], who introduced Graph Deviation Networks (GDN) for multivariate time series. GDN learns graph structure from training data and performs node-wise anomaly detection by predicting deviations from normal behaviour. Each node aggregates information from its neighbours via attention-based graph convolution and forecasts its next value. Anomalies are detected by comparing the predicted and actual values. This model showed that incorporating structural priors via learned graphs leads to significant improvements over unstructured forecasting methods.

Another influential contribution is from Zhao et al. [62], who proposed MTAD-GAT, a Multi-Task Anomaly Detection model using Graph Attention Networks. MTAD-GAT builds two types of attention mechanisms: one for capturing spatial dependencies across

sensors, and the other for modelling temporal dependencies within each sensor. The multi-task setup includes both forecasting and reconstruction tasks, encouraging the model to learn a more robust representation of normal patterns. This dual-attention architecture demonstrated superior performance on several benchmarks, highlighting the effectiveness of integrating GNNs into anomaly detection pipelines.

GNNs also provide flexibility in terms of graph construction. While some models rely on prior knowledge of sensor relationships (e.g., physical proximity or wiring diagrams), others adopt data-driven approaches to learn adjacency matrices from statistical correlations or attention mechanisms. For example, the work by Chen et al. [15] integrates Transformer encoders and self-attention to learn latent graph structures adaptively. This method captures complex and evolving inter-sensor relationships, which are often difficult to encode manually.

Beyond temporal modelling, GNNs have been extended to model more complex forms of interaction. Wu et al. [54] present a comprehensive survey on recent advancements where heterogeneous graphs, hypergraphs, and dynamic graphs are used to capture multi-dimensional relationships. These innovations enable GNN-based models to go beyond static and homogeneous representations, making them suitable for anomaly detection in evolving and heterogeneous environments [23].

In practical deployments, GNN-based anomaly detectors have proven effective in detecting both point anomalies and contextual anomalies. For example, Hundman et al. [29] applied a variant of graph-based forecasting for NASA's telemetry data, showing that structurally informed models outperform traditional univariate or fully connected models. Moreover, recent industrial applications such as IT infrastructure monitoring and predictive maintenance increasingly incorporate GNNs to enhance anomaly detection accuracy while reducing false alarms [36].

Despite their success, GNN-based approaches face several challenges. First, the quality of the learned graph structure significantly affects performance. Overly sparse or dense graphs can hinder learning and generalisation [55]. Second, scalability remains an issue in very large sensor networks, where message passing can become computationally expensive [52]. Finally, interpretability of GNN models is still limited, although recent efforts in explainable GNNs (e.g., GNN Explainer [58]) have made some progress.

To summarise, GNNs offer a principled framework for incorporating spatial and contextual dependencies into anomaly detection models. Their ability to learn and reason over structured data aligns naturally with the characteristics of multivariate time series from sensor networks. As research continues to advance in graph learning and model

interpretability, GNN-based methods are expected to play an increasingly central role in robust and scalable anomaly detection systems [14].

2.4 Spatial Temporal Graph Neural Networks for Anomaly Detection

STGNNs have emerged as a powerful framework for modelling multivariate time-series data where both spatial and temporal dependencies are present. In ICS, sensors are often deployed across diverse physical components, each producing temporally correlated streams of data. The temporal evolution of each sensor reading and the spatial correlation between sensors are both essential for accurate anomaly detection. Traditional deep learning models, such as convolutional or recurrent neural networks, are capable of capturing temporal dynamics but fail to model complex spatial relationships unless explicitly engineered. STGNNs address this limitation by fusing graph-based spatial modelling with temporal sequence processing, making them particularly suitable for anomaly detection in sensor networks [53, 63].

The central idea of STGNNs is to represent the sensor network as a graph $G = (V, E)$, where each node $v \in V$ corresponds to a sensor and edges E represent physical or learned relationships. Temporal signals are then propagated across this graph using both message-passing operations and temporal encoders. This dual-structure allows the model to learn dynamic correlations not just within individual time-series but also among interconnected sensors, enabling the detection of both point anomalies and context-dependent deviations [56].

One of the foundational models in this domain is the Spatial Temporal Graph Convolutional Network (ST-GCN) introduced by Yan et al. [56], which was initially developed for skeleton-based action recognition. Though not designed for anomaly detection, it inspired subsequent adaptations where time-series data are treated as sequences of graph signals. ST-GCN employs spatial graph convolutions to capture inter-node relationships and temporal convolutions to model intra-node dynamics, forming the basis for many subsequent STGNN variants.

In the context of anomaly detection, several works have extended this architecture to better suit multivariate time-series data. Wu et al. [53] proposed the Graph WaveNet model, which uses adaptive adjacency matrices and dilated temporal convolutions to model both short-range and long-range dependencies. The adaptive graph learning component allows the model to infer latent spatial structures from data, while temporal

convolutions improve efficiency and allow parallel training. Although originally applied to traffic forecasting, Graph WaveNet's principles have been adapted for anomaly detection in industrial sensor networks [63].

To explicitly address anomaly detection, Deng and Hooi [18] proposed GDN, which include temporal encoders and attention-based graph aggregation layers. Though not classified as an STGNN, GDN's design contains elements of both spatial and temporal modelling. The forecasting-based detection approach in GDN inspired further work that integrates spatial temporal patterns more cohesively [62].

Recent research has focused on enhancing the flexibility and interpretability of STGNNs for anomaly detection. For instance, Kim et al. [31] introduced STG-NAD (Spatial Temporal Graph Neural Network for Anomaly Detection), which fuses graph attention mechanisms with temporal convolutional networks to localise anomalies in space and time. Their model demonstrates that by localising anomalous influence in both graph and time domains, detection performance can be significantly improved while maintaining interpretability.

Other approaches, such as CST-GNN (Correlation-aware Spatial-Temporal Graph Neural Network) by Zheng et al. [63], enhance spatial modelling by masking connections across dissimilar sensor clusters, enabling the model to better focus on meaningful structural relationships. CST-GNN further integrates a PCA-based scoring mechanism, which isolates anomaly-indicative variations in the forecast error space, making it robust to noise and feature drift. The use of cluster-constrained graphs and directional message passing marks a novel step forward in structurally aware anomaly detection [15].

A notable advantage of STGNNs is their ability to model long-range dependencies that arise both spatially and temporally. Unlike recurrent models, which suffer from vanishing gradients and require sequential processing, STGNNs leverage temporal convolution and residual connections, allowing for deeper and more scalable architectures [53]. Additionally, graph-based inductive biases reduce the burden of learning from scratch, allowing models to generalise better even with limited anomaly samples [63].

In terms of practical applications, STGNNs have shown strong performance across various domains. For instance, Su et al. [49] evaluated several STGNN-based models on the Server Machine Dataset (SMD), demonstrating superior accuracy and lower false positive rates compared to conventional baselines. In smart grid monitoring and water treatment systems, STGNNs enable real-time fault detection, minimising downtime and maintenance costs [3, 43].

Nevertheless, several challenges remain. One open issue is the determination of

an appropriate graph structure. While adaptive methods allow the model to learn the graph end-to-end, they may introduce noise or spurious connections without proper regularisation [55]. Additionally, real-world deployment demands robustness to sensor failure, data noise, and latency constraints, which current models are only partially equipped to handle [36]. Addressing these concerns requires better integration of graph learning with domain knowledge, as well as advances in explain-ability and uncertainty estimation [58].

In conclusion, Spatial Temporal GNNs represent a significant advance in anomaly detection methodologies for multivariate time-series data. By integrating spatial context and temporal dynamics within a unified neural framework, they provide superior capacity to model complex dependencies inherent in ICS and related systems. As more scalable and interpretable variants continue to emerge, STGNNs are poised to become a foundational approach for trustworthy and accurate anomaly detection in large-scale, sensor-driven environments [52, 63].

2.5 Datasets for Multivariate Time-Series Anomaly Detection

A central component of evaluating MTSAD models is the choice of datasets. The quality, complexity, and structure of datasets directly influence the development, benchmarking, and generalisation of new anomaly detection methodologies. Among the available options, two datasets have emerged as standard benchmarks within the MTSAD and ICS research community: SWaT (Secure Water Treatment) and WADI (Water Distribution). This section provides an overview of these datasets, discusses their relevance and strengths, and compares them with alternative datasets in the field.

The SWaT dataset, introduced by Mathur and Tippenhauer [43], was collected from a scaled-down version of a real-world water treatment plant located in Singapore. The testbed includes multiple stages of water treatment such as chemical dosing, sedimentation, filtration, and reverse osmosis. It contains 51 attributes comprising sensor readings (e.g., flow rate, pH, level) and actuator states (e.g., pump on/off). The dataset spans 11 days: 7 days of normal operation and 4 days with labelled attack scenarios. Each attack targets different components of the system and simulates cyber-physical attacks such as actuator override or sensor spoofing. SWaT's detailed attack annotations, realistic process control dynamics, and multi-stage architecture make it an ideal benchmark for evaluating MTSAD models under both temporal and structural anomalies.

The WADI dataset [3] was collected from an extended version of the SWaT system, designed to simulate a full-scale water distribution network. It incorporates a larger number of sensors and control loops, increasing the system's complexity and heterogeneity. WADI contains 16 days of data: 14 days of normal operation and 2 days with simulated attacks. The attacks cover diverse anomaly types, including water leakage, backflow, and control logic tampering. Compared to SWaT, WADI offers higher-dimensional data, broader subsystem coverage, and more complex temporal correlations among sensor groups. This complexity makes it especially suitable for testing models that incorporate spatial-temporal modelling, sensor clustering, and adaptive graph learning.

One of the major reasons for focusing on SWaT and WADI in this thesis is their alignment with the core challenges addressed by my proposed framework. Both datasets exhibit modular physical architectures where sensors within the same stage are more strongly correlated than across stages. This modularity motivates the use of clustering techniques and cluster-constrained graph neural networks. Moreover, the availability of fine-grained ground truth labels for attack events allows for robust supervised evaluation using precision, recall, and AUC metrics.

Compared to other public datasets such as SMD (Server Machine Dataset) [24], NAB [33], and Yahoo Webscope [32], SWaT and WADI offer several advantages. First, they originate from real-world industrial control systems, thereby capturing operational noise, hardware constraints, and actuator dependencies not present in synthetic datasets. Second, they contain both continuous and discrete signals, necessitating hybrid modelling approaches. Third, their annotated attack periods enable ground truth-based quantitative evaluation, unlike many unsupervised datasets where anomaly labels are incomplete or subjective.

To provide a quantitative overview and facilitate comparison, Table 2.1 summarizes key statistics of the SWaT, WADI, and other commonly used benchmark datasets such as SMD. The metrics include the number of channels (sensors/features), the number of samples in the training and testing sets, and the percentage of anomalous samples in the test set.

As shown in Table 2.1, SWaT and WADI, while having fewer total samples than SMD, exhibit a higher proportion of anomalies, especially SWaT. This reflects the realistic nature of the attack scenarios captured in these ICS testbeds. Furthermore, the higher dimensionality of WADI (127 channels) presents a more significant challenge for modelling inter-sensor relationships compared to SWaT (51 channels) and SMD (38 channels).

Table 2.1: Statistics of benchmark datasets for multivariate time-series anomaly detection.

Dataset	# Channels	# Train	# Test	% Anomalies	Domain
SWaT	51	47,515	44,986	11.97%	Water Treatment
WADI	127	118,795	17,275	5.99%	Water Treatment
SMD	38	304,168	304,174	5.84%	Server Infrastructure
NAB	1–3	21,000	21,000	1.10%	KPI Streams
Yahoo Webscope	1	67,000	67,000	1.56%	Web Traffic

The SMD [49] is often cited for its high-dimensional time series data, with each server generating 38 features. However, prior studies [36] have noted that SMD suffers from severe concept drift in many of its machines, rendering a significant portion of its anomalies hard to validate. This limits its reliability for structured experimental comparison. Similarly, Yahoo Webscope and NAB are more appropriate for low-dimensional or point-wise anomalies and less applicable to structured anomaly patterns in ICS.

By focusing on SWaT and WADI, this work targets realistic, multi-sensor, modular systems with operational heterogeneity. These datasets also reflect practical deployment challenges in anomaly detection: complex causality, inter-variable noise, and simultaneous faults. They provide a suitable testbed for validating advanced modelling strategies such as spatial temporal graph learning and multi-task forecasting.

2.6 Summary

This chapter has examined the current landscape of multivariate time-series anomaly detection, emphasising sensor clustering and graph-based modelling techniques. I first surveyed general anomaly detection methodologies, then explored the rationale and techniques behind sensor clustering. Next, I reviewed recent advances in graph neural networks and spatial-temporal extensions for anomaly detection, identifying the evolution from simple correlation structures to cluster-aware spatial-temporal learning. Finally, I justified my dataset choice by analysing the structural and operational characteristics of SWaT and WADI. Together, these components form a cohesive foundation for the design of the MLAD (Multi-task Learning for Anomaly Detection) framework presented in the next chapters.

CHAPTER 3

SENSOR EMBEDDING AND CLUSTERING

In multivariate time-series anomaly detection tasks, sensors often exhibit significant behavioural differences due to structural heterogeneity and functional diversity. Grouping all sensors into a unified modelling framework may result in the suppression of local anomalies and instability in predictions, thereby compromising both accuracy and interpretability.

To address these issues, this chapter introduces an unsupervised approach for sensor behaviour embedding and clustering. The goal is to extract temporal patterns from each sensor and group sensors with similar dynamics. The resulting clusters are later used to define spatial structures for graph learning and modular prediction. I employ UMAP (Uniform Manifold Approximation and Projection) [44] for structure-preserving dimensionality reduction, followed by DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [47] for clustering sensors based on their behavioural embeddings. To support my design choices, I also compare DBSCAN with K-Means and analyse the impact of clustering through ablation studies.

The following sections describe the full implementation process, present visual results, and analyse the effectiveness and interpretability of this clustering approach within the MLAD (Multi-task Learning for Anomaly Detection) framework.

3.1 Data Preprocessing and Feature Embedding

To perform sensor clustering effectively, I first extract representative statistical features that reflect each sensor's temporal behaviour. For both the SWaT and WADI datasets, a consistent data preprocessing procedure is applied to construct feature representations for each sensor.

Let the original multivariate time-series data be represented as:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]^\top \in \mathbb{R}^{T \times N} \tag{3.1}$$

where T is the number of time steps, N is the number of sensors, and $\mathbf{x}_t \in \mathbb{R}^N$ denotes the sensor readings at time t.

A fixed-size sliding window is applied to each sensor dimension i to obtain a sequence of local time segments:

$$\mathscr{S}_i = \left\{ \mathbf{x}^{(i)}_{t:t+w-1} \mid t = 1, 1+s, 1+2s, \ldots \right\}, \quad \mathbf{x}^{(i)}_{t:t+w-1} \in \mathbb{R}^w \tag{3.2}$$

where w is the window size and s is the step size.

For each segment, I compute a set of features, including mean, variance, maximum, minimum, median, skewness, and kurtosis, which are computed to capture the distributional characteristics of the sensor readings. These are then aggregated and standardised to form a feature vector $\mathbf{e}_i \in \mathbb{R}^d$ for sensor i:

$$\mathbf{e}_i = \text{Aggregate}(\mathscr{S}_i), \quad i = 1, \ldots, N \tag{3.3}$$

The collection of all sensor feature vectors constitutes the high-dimensional feature space describing system-wide operational behaviour. This feature space extracted entirely from normal data serves as the input for the subsequent UMAP-based dimensionality reduction, enabling unsupervised grouping of behaviourally similar sensors.

To illustrate the motivation for sensor grouping, Figure 3.1 presents three representative behavioural patterns observed across sensors in the SWaT and WADI datasets. For example, Pattern 1 shows strong periodicity, Pattern 2 demonstrates gradual drift, while Pattern 3 contains abrupt shifts or sharp local variations. These examples highlight the intrinsic heterogeneity in sensor behaviours, indicating that it is unreasonable to treat all sensor dimensions equally in a unified model.

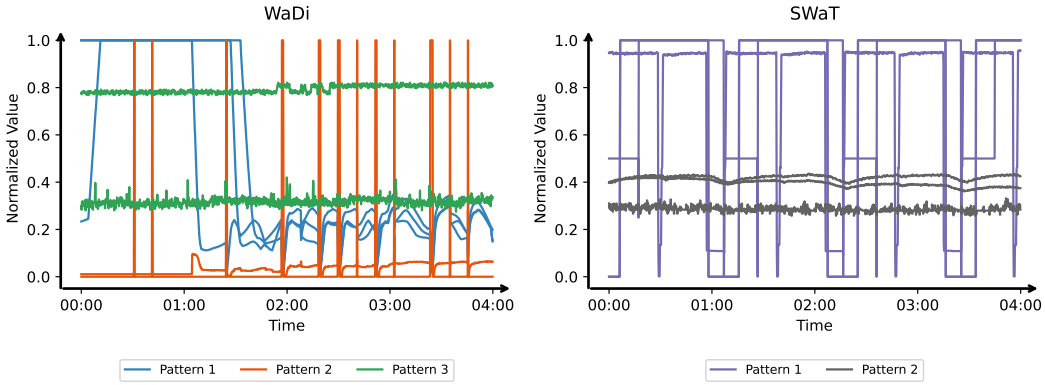

Figure 3.1: Examples of sensor behaviour patterns in the WADI (left) and SWaT (right) datasets. Sensors exhibit diverse dynamics such as oscillation, drift, or sudden transitions, which motivate unsupervised grouping.

3.2 Introduction to UMAP

Uniform Manifold Approximation and Projection (UMAP) is a nonlinear dimensionality reduction technique that is particularly effective at preserving the local structure of high-dimensional data. Unlike PCA (Principal Component Analysis) [48], which is linear, and t-SNE (t-distributed Stochastic Neighbor Embedding) [41], which prioritises neighbourhood clustering but distorts global distances, UMAP offers a balance between local fidelity and global geometry.

UMAP operates on the assumption that data lie on a Riemannian manifold and attempts to preserve the topological structure by minimising the cross-entropy between local fuzzy simplicial sets in high and low dimensions. Its objective function aims to minimise the following divergence between pairwise probabilities:

$$\mathscr{L}_{UMAP} = \sum_{i \neq j} w_{ij} \cdot \log\left(\frac{w_{ij}}{q_{ij}}\right)$$

Here, w_{ij} represents the pairwise similarity in the high-dimensional space and q_{ij} the corresponding similarity in the embedded space.

Compared to t-SNE, UMAP scales better to large datasets and supports deterministic embeddings, which is beneficial for reproducibility. In this work, I apply UMAP to compress the extracted statistical feature embeddings from each sensor into a 2D latent space while retaining their neighbourhood structure, thus making the clustering process more effective and interpretable.

The choice of UMAP is particularly appropriate for the characteristics of industrial multi-sensor data used in this study. Sensor behaviour features are typically high-dimensional and exhibit nonlinear relationships caused by physical coupling between system components. For example, variables such as flow rate, pressure, and valve states often evolve together due to underlying control mechanisms. These dependencies imply that the observed high-dimensional feature vectors are not uniformly distributed in Euclidean space but tend to lie on a lower-dimensional manifold governed by the physical dynamics of the system.

This observation aligns well with the fundamental assumption of UMAP, which assumes that high-dimensional observations are sampled from a locally connected Riemannian manifold. By preserving neighbourhood relationships within this manifold, UMAP is able to capture the intrinsic structure of sensor behaviour patterns. In contrast, linear dimensionality reduction methods such as PCA cannot model nonlinear dependencies, while t-SNE focuses primarily on local clustering and may distort the global structure of the data.

Furthermore, UMAP provides better scalability and stability when applied to large sensor datasets such as SWaT and WADI, where hundreds of sensors produce heterogeneous behaviour patterns. These properties make UMAP particularly suitable for constructing behaviour embeddings that preserve meaningful relationships among sensors before the clustering stage.

3.3 Clustering Method Selection and DBSCAN Implementation

After obtaining sensor behaviour embeddings through dimensionality reduction, I adopt the density-based clustering method DBSCAN to group the sensors in an unsupervised manner. DBSCAN is capable of identifying clusters with arbitrary shapes and detecting outliers without requiring the number of clusters to be specified in advance. This makes it particularly suitable for CPS (cyber-physical systems) scenarios where sensor distributions are complex and the number of behavioural groups is unknown.

The core principle of DBSCAN is to construct clusters based on density connectivity, controlled by two hyper-parameters: neighbourhood radius ε and minimum neighbours $MinPts$. A point is considered a *core point* if its ε neighbourhood contains at least $MinPts$ other points. Points within a core point's neighbourhood are considered to belong to the same cluster. In my experiments, I empirically set $\varepsilon = 0.25$ and $MinPts = 3$ for

both the SWaT and WADI datasets, which resulted in stable and meaningful clustering structures. DBSCAN not only adapts well to the true distribution of sensor behaviours but also naturally identifies anomalous or irregular sensors as outliers, which supports downstream anomaly detection.

The suitability of DBSCAN for this task is closely related to the intrinsic distribution characteristics of sensor behaviour embeddings. In industrial cyber-physical systems, sensors are deployed to monitor different physical components and processes, which leads to heterogeneous behavioural patterns across sensors. After the UMAP embedding step, sensors with similar temporal dynamics tend to form locally dense regions in the embedding space, while sensors with distinct behaviours appear in sparse regions.

This density-based structure naturally matches the clustering assumption of DBSCAN, which identifies clusters as regions with high point density separated by areas of low density. Unlike centroid-based methods such as K-Means, DBSCAN does not require clusters to follow spherical or convex shapes. This is particularly important in sensor behaviour embeddings, where clusters may exhibit irregular or elongated structures due to different operational regimes or physical coupling between sensors.

Furthermore, DBSCAN can automatically identify noise points that do not belong to any dense region. In the context of sensor networks, such points often correspond to sensors with unique or unstable behaviour patterns. Identifying these outliers is beneficial for anomaly detection, as these sensors may require special modelling or may represent potential fault sources. Therefore, DBSCAN provides a flexible and robust clustering mechanism for discovering the true distribution of sensor behaviour patterns in the embedded feature space.

To validate the effectiveness of DBSCAN, I compared it against the classic K-Means clustering algorithm. While K-Means is widely used and performs well when data exhibits regular and spherical cluster shapes, it has limitations when applied to multivariate time-series sensor embeddings. First, K-Means relies heavily on Euclidean distance, which is inadequate for capturing similarity between global temporal behaviour patterns. Second, sensor embeddings in my scenario often form non-convex, irregular clusters due to varying sampling rates and functional roles. This violates K-Means isotropic assumption. Additionally, K-Means requires a predefined cluster count k, which is impractical when prior knowledge is lacking. Its sensitivity to initialisation also results in unstable clustering outcomes.

Empirically, I observe that K-Means tends to misgroup sensors with dissimilar physical roles into the same cluster, or split highly similar sensors across clusters. These

misgroupings degrade the interpretability of the resulting graph structure and affect the modular design of subsequent prediction networks.

To further quantify the impact of the two clustering approaches on model performance, I implemented two variants of the MLAD model using DBSCAN and K-Means respectively, under otherwise identical conditions. Evaluation on the SWaT and WADI datasets reveals that the DBSCAN-based model consistently outperforms its K-Means counterpart in both ROC AUC and PRC AUC metrics. As shown in Figure 3.2, DBSCAN yields a 1.6% and 2.4% improvement on ROC AUC and PRC AUC respectively on SWaT, and a 3.3% and 2.8% improvement on WADI. These results confirm the suitability of DBSCAN for behavioural clustering in high-dimensional time-series settings.

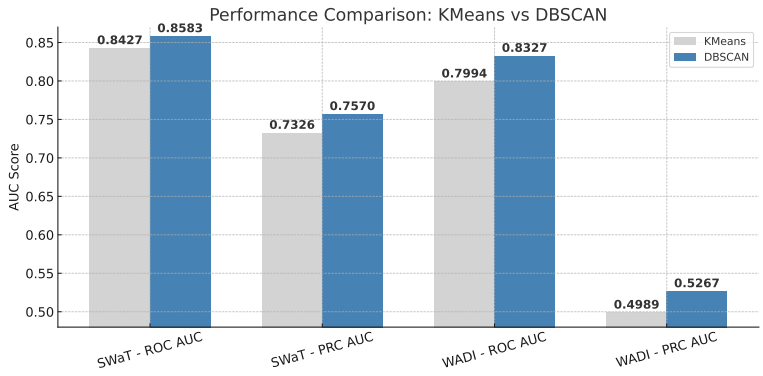

Figure 3.2: Performance comparison of the MLAD model using K-Means and DBSCAN clustering on the SWaT and WADI datasets. DBSCAN consistently outperforms K-Means, demonstrating its robustness in modelling complex sensor behaviour.

Finally, the clustering result is used to define subgraph partitions over sensors, forming the basis for graph-based modelling in subsequent modules. Each cluster corresponds to a group of functionally related sensors, enabling modular design in the multi-task forecasting architecture.

3.4 Alignment Between Clusters and Physical System Structure

Remarkably, although the clustering is based purely on temporal similarity, the results exhibit strong alignment with known physical subsystems. For example, in SWaT, DBSCAN naturally groups sensors from the same processing stages such as pumps, valves, or chemical tanks. In WADI, even with more complex layouts, clustering results tend to

correspond to nearby devices or closely coupled processes. This structure implies that my method captures latent functional organisation.

Figure 3.3 shows how clusters align with the physical layout of the SWaT system. Most clusters map to a specific functional block.

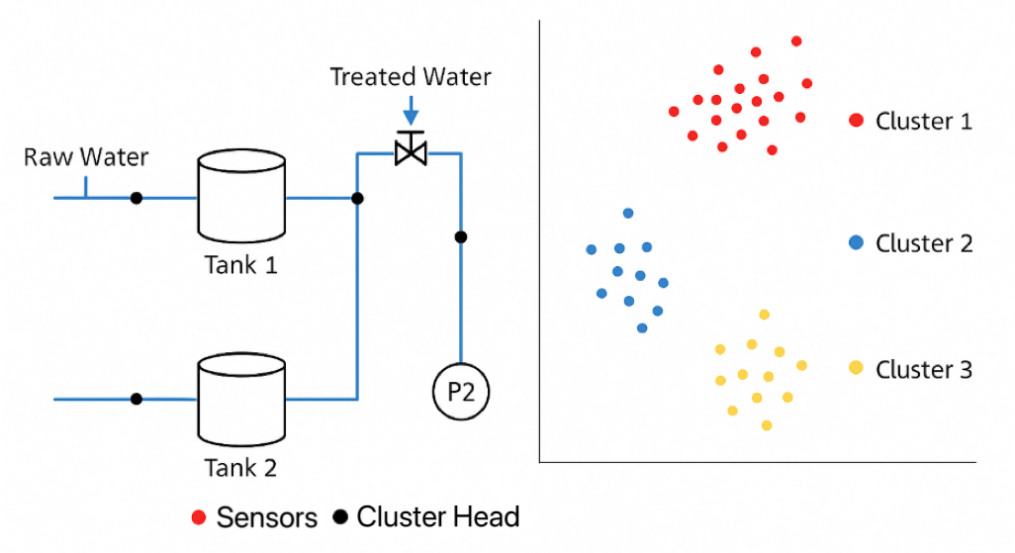

Figure 3.3: Mapping between clustered sensor groups and physical system layout. Left: simplified SWaT layout with sensors grouped around Tank 1, Tank 2, and Pump P2. Right: DBSCAN clusters in UMAP-embedded space.

3.5 UMAP-Based Temporal Embedding

To structure sensor behaviour more effectively, I apply UMAP to reduce the high-dimensional feature matrix $\mathbf{E}$ into a low-dimensional latent space:

$$\mathbf{Z} = \text{UMAP}(\mathbf{E}) \in \mathbb{R}^{N \times 2} \tag{3.4}$$

UMAP preserves local relationships and density in the feature space. It minimises the divergence between pairwise similarity distributions in high and low dimensions:

$$\min_{Y} \quad \mathscr{L}_{\text{UMAP}} = \sum_{i \neq j} w_{ij} \cdot \log\left(\frac{w_{ij}}{q_{ij}}\right) \tag{3.5}$$

where $Y \in \mathbb{R}^{N \times d'}$ represents the low-dimensional embedding coordinates of the N sensors that UMAP learns during optimisation, and w_{ij} and q_{ij} denote the similarity weights in the high-dimensional feature space and the embedded space, respectively.

For sensor clustering, the UMAP embedding dimension is set to 5, and applied DBSCAN with an epsilon of 0.25 and a minimum samples threshold of 3 to identify sensor clusters.The number of neighbours is set to 15, 30 and 10 for, and the minimum distance parameter is 0.1. This enables meaningful separation among sensor clustering.

3.6 Evaluation and Ablation Analysis

I evaluate the impact of clustering by comparing against two baselines: one without clustering (all sensors modelled together), and one with clustering. On both SWaT and WADI, the full model with DBSCAN clustering achieves significantly lower forecasting errors and better anomaly detection scores.

On SWaT, the ROC AUC improves from 0.8446 (no clustering) to 0.8583 (with clustering) and the PRC AUC improves from 0.7471 (no clustering) to 0.7570 (with clustering). On WADI, the ROC AUC improves from 0.7984 (no clustering) to 0.8327 (with clustering) and the PRC AUC improves from 0.5154 (no clustering) to 0.5267 (with clustering). Ablation results confirm that meaningful sensor grouping improves both learning stability and detection robustness.

Figure 3.4 summarises the comparative performance across methods.

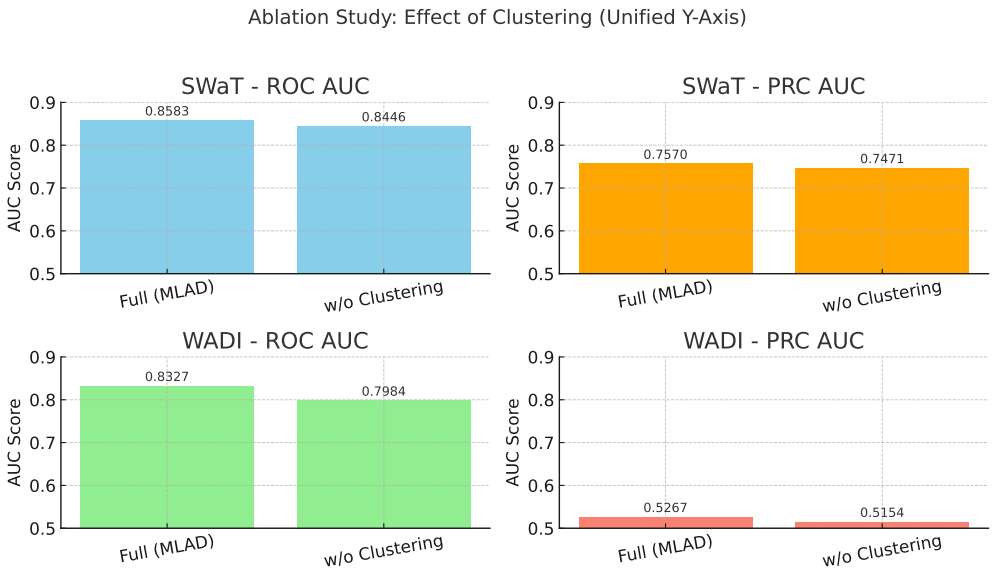

Figure 3.4: Ablation study results comparing the full model (MLAD) and its variant without the clustering module. The full version consistently outperforms in both ROC AUC and PRC AUC across SWaT and WADI datasets.

3.7 Summary

This chapter presented the design and implementation of the sensor embedding and clustering module in the MLAD framework. I proposed an unsupervised, data-driven approach that starts with statistical feature extraction over sliding windows, followed by UMAP-based dimensionality reduction and DBSCAN clustering.

Experimental results demonstrate that DBSCAN is more suitable than K-Means for SWaT and WADI datasets, which contain complex temporal dynamics and irregular cluster shapes. The resulting clusters not only reflect functional similarities among sensors but also align with physical subsystem layouts.

Ablation studies confirm that clustering plays a critical role in improving forecasting accuracy and anomaly detection performance. Compared to baselines without clustering, the full model achieves significantly better results, showing higher robustness and generalisation.

Overall, the clustering module lays the foundation for structural decomposition and interpretability. In the next chapter, I leverage these clusters to construct graph structures and design a spatially constrained GNN for predictive anomaly detection.

CHAPTER 4

SPATIAL TEMPORAL GNN MODELLING

Building upon the sensor clustering results discussed in the previous chapter, this chapter presents the core model proposed in this study: MLAD (Multi-task Learning for Anomaly Detection), a spatial-temporal graph neural network-based model designed for accurate and robust multivariate time series anomaly detection. While traditional sequence models such as LSTM (Long Short-Term Memory) [29] or GRU (Gated Recurrent Unit) [19] are effective at capturing temporal dependencies, they typically fail to model the complex spatial relationships between sensors, especially when these relationships do not stem from physical layouts but from implicit behavioural similarities. Conversely, conventional GNN-based models (e.g., ST-GCN [57], DCRNN [35]) rely heavily on predefined or static graph structures, often built from physical adjacency or Euclidean distances. These approaches struggle to reflect the functional similarity of sensors, limiting their effectiveness in systems without a clear spatial prior.

To address these challenges, MLAD introduces a behaviour-based clustering strategy as the foundation for constructing graph structures. This enables the model to define sensor connections based on dynamic similarity rather than physical proximity, introducing flexibility, adaptability, and improved task alignment compared to static GNN models. The architecture of MLAD comprises three main components:

- A clustering-based graph construction module that generates sparse and meaningful adjacency matrices;
- A spatial encoding module using Graph Convolutional Networks (GCN) to model

inter-sensor relations at each time step;

- A temporal modelling module using Temporal Convolutional Networks (TCN) to capture sequential dynamics.

Furthermore, MLAD adopts a multi-task learning strategy with two prediction objectives forecasting future values and reconstructing past inputs. This dual-path design strengthens the model's capacity to learn both trend dynamics and stable patterns, thereby enhancing its ability to detect a diverse range of anomalies, from abrupt spikes to gradual deviations.

This chapter details the specific design of the MLAD model, including the GCN and TCN modules, loss functions, and the associated training procedures. An overview of the entire framework is first provided to set the stage for subsequent technical elaboration.

4.1 Overview of the Framework

An overview of the MLAD model is shown in Figure 4.1. The central idea is to use sensor clustering to define spatial groupings, followed by spatial graph modelling and temporal sequence prediction, jointly optimised through a multi-task loss.

The model input is a sliding window of length T, covering sensor observations from T previous time steps. Each observation vector $\mathbf{x}_t \in \mathbb{R}^N$ contains the values from N sensors. After normalisation, the data is passed into the spatial graph encoder.

During spatial modelling, the model constructs a multi-subgraph structure based on the clustering result from Chapter 3. Each cluster defines a fully connected subgraph. The GCN is applied to extract spatial features at each time step. The spatial representations across time are then fed into a TCN, which captures the temporal evolution of sensor states and makes predictions for future values.

To enhance the learning of both temporal stability and evolving trends, MLAD introduces two parallel prediction heads:

- A reconstruction head, which reconstructs the original inputs within the sliding window;

- A forecasting head, which predicts one or more steps ahead.

The final loss is computed as a weighted sum of the reconstruction and forecasting losses, allowing the model to detect a wide range of anomaly types.

This section provides a high-level understanding of the MLAD architecture. The following sections will detail the specific design of each component, including cluster-constrained graph construction, the GCN and TCN modules.

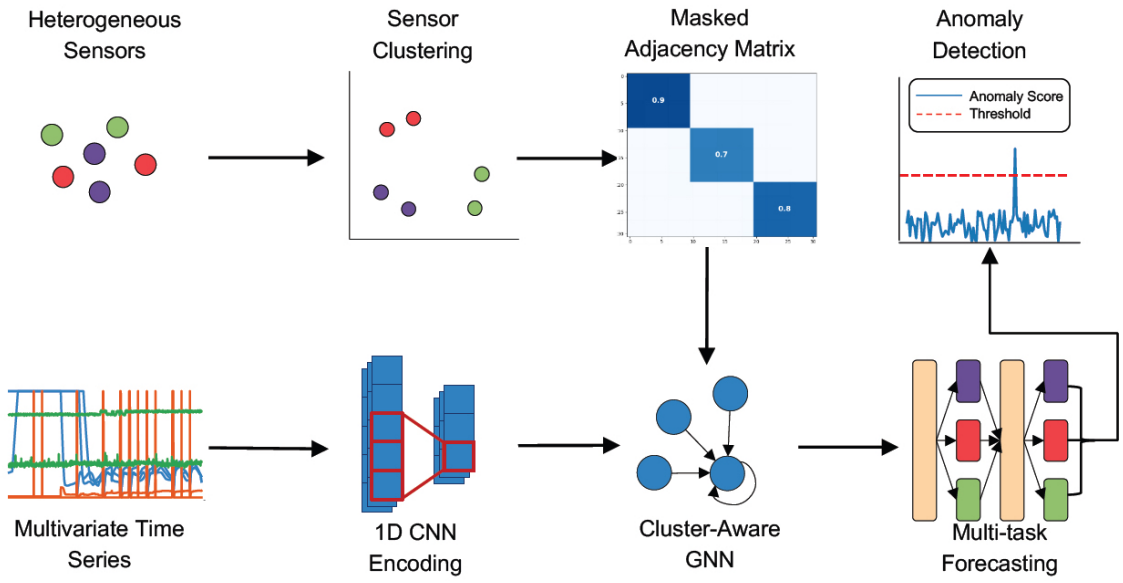

Figure 4.1: Overall architecture of the MLAD model. The framework consists of a clustering-based graph encoder, a temporal convolutional module, and dual-task output heads.

4.2 Clustering Graph Construction

In MLAD, the core objective of spatial graph construction is to define the connectivity structure among sensors that is, to determine how information flows between nodes in the graph. Unlike traditional GNNs that rely on physical adjacency or manually specified graphs, MLAD adopts a cluster-driven approach that emphasises behavioural similarity over physical proximity.

Specifically, the DBSCAN-based clustering process from Chapter 3 groups sensors into behaviourally similar clusters. For each cluster, a fully connected subgraph is created in which all nodes (sensors) are linked to each other, assuming potential behavioural correlations. The union of all such subgraphs forms the complete graph used by the GCN.

This strategy provides several advantages:

- It is data-driven and adaptable to different datasets and system architectures without needing prior physical knowledge;

- It preserves local sensor relationships that are functionally consistent, improving the model's sensitivity to local anomalies;

- It yields a sparse and structured adjacency matrix, which improves computational efficiency.

The graph is represented by an adjacency matrix $A \in \mathbb{R}^{N \times N}$, where N is the number of sensors. The entries are defined as:

$$A_{ij} = \begin{cases} 1, & \text{if sensor } i \text{ and } j \text{ are in the same cluster,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

This adjacency matrix acts as a structural mask that constrains spatial information propagation during the GCN operations.

Figure 4.2 illustrates an example of the resulting cluster mask. Distinct block patterns emerge, each corresponding to a separate behavioural group. This structure ensures that the spatial modelling process focuses on sensors with similar behaviour, enhancing both stability and interpretability.

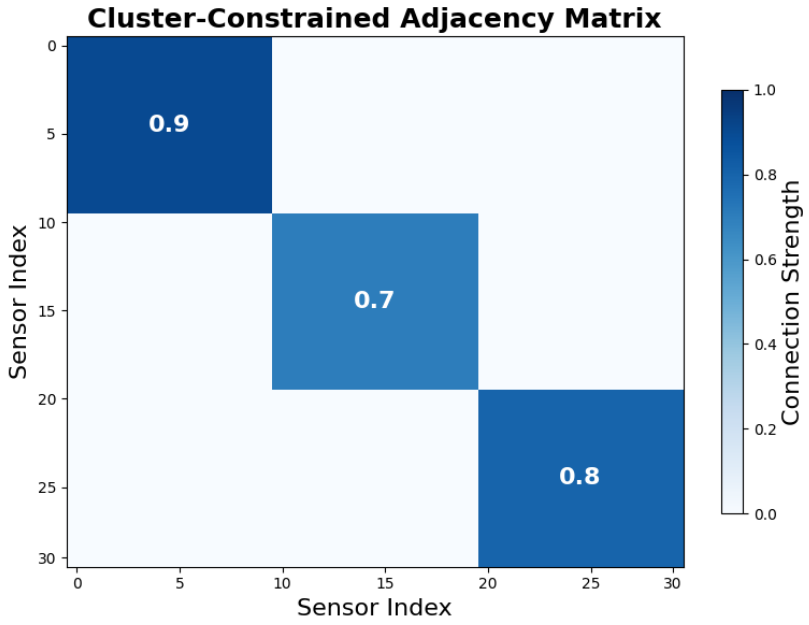

Figure 4.2: Example of a cluster-constrained adjacency matrix used in MLAD. Each block corresponds to a fully connected subgraph of behaviourally similar sensors.

4.3 Spatial Dependency Encoding via GCN

After constructing the sensor graph using cluster-based adjacency, the MLAD model employs Graph Convolutional Networks (GCNs) to capture spatial dependencies among sensors. GCNs enable each sensor node to aggregate features from behaviourally similar nodes, as defined by the cluster-constrained adjacency matrix.

Let $\mathbf{X}_t \in \mathbb{R}^{N \times d_{in}}$ denote the sensor input features at time step t, where N is the number of sensors and d_{in} is the input dimension. The spatial encoding is performed by applying a two-layer GCN as follows:

$$\mathbf{H}_t^{(1)} = \mathrm{ReLU}(\hat{A}\mathbf{X}_t\mathbf{W}^{(1)}), \tag{4.2}$$

$$\mathbf{H}_t^{(2)} = \hat{A}\mathbf{H}_t^{(1)}\mathbf{W}^{(2)} \tag{4.3}$$

where $\hat{A}$ is the normalised adjacency matrix with self-connections, and $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ are learnable weight matrices. The first layer applies a non-linear transformation with ReLU, while the second layer outputs the final spatial representation $\mathbf{H}_t = \mathbf{H}_t^{(2)}$.

This formulation allows the model to integrate information from neighbouring sensors within the same cluster, effectively capturing functional dependencies without relying on physical topology. Since each cluster defines a fully connected subgraph, the GCN can capture high-order correlations among temporally aligned sensors. The two-layer structure strikes a balance between expressive power and computational cost, enabling efficient training while maintaining sufficient spatial resolution. By using shared parameters across all time steps, the GCN layer generalises well and reduces overfitting. The output $\mathbf{H}t \in \mathbb{R}^{N \times d_{out}}$ is a set of spatially enhanced sensor features, which are then passed to the temporal modelling module described in the next section. Figure 4.3 illustrates the concept of spatial feature propagation using a GCN layer applied within a clustered sensor graph.

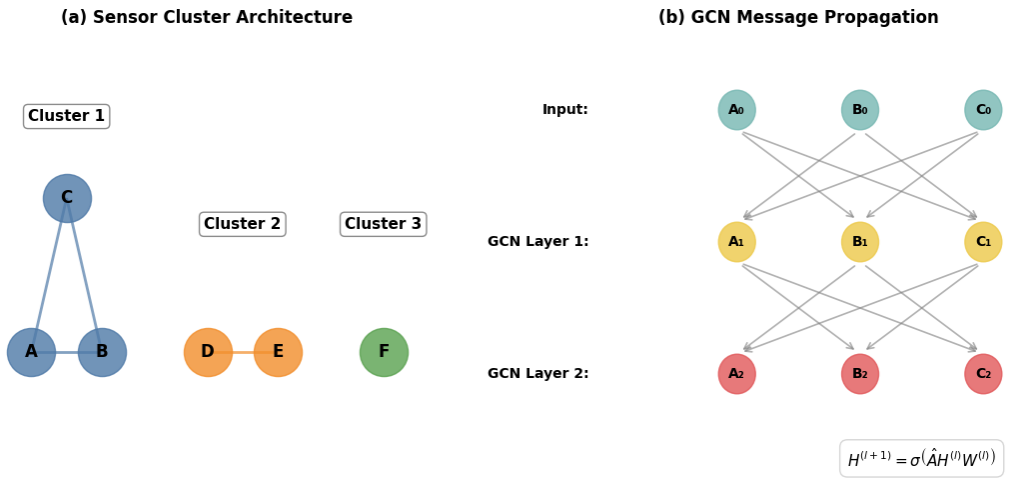

Figure 4.3: GCN-based spatial feature propagation within sensor clusters. (**a**) Cluster architecture showing three distinct sensor groups with intra-cluster connectivity. (**b**) Two-layer GCN message passing process demonstrating feature aggregation from neighbouring sensors within the same cluster. The mathematical formulation illustrates the layer-wise propagation rule.

4.4 Temporal Dependency Encoding via TCN

Following the extraction of spatial features using GCN, MLAD employs a Temporal Convolutional Network (TCN) to model the sequential evolution of sensor states over time. Unlike recurrent models such as LSTM or GRU, TCN allows parallel computation across time steps, improved stability during training, and larger receptive fields with fewer layers.

Let the spatially encoded features over a sliding window of length W be denoted as $\{\mathbf{H}_{t-W+1}, \mathbf{H}_{t-W+2}, \ldots, \mathbf{H}_t\}$, where each $\mathbf{H}_t \in \mathbb{R}^{N \times d}$ is the spatial representation at time t. These feature matrices are stacked along the temporal dimension to form a sequence tensor suitable for temporal processing:

$$\mathscr{H} = \text{stack}(\mathbf{H}_{t-W+1}, \mathbf{H}_{t-W+2}, \ldots, \mathbf{H}_t) \in \mathbb{R}^{W \times N \times d} \tag{4.4}$$

To model temporal dynamics, a shared TCN is applied across all sensor nodes. The input tensor is processed along the time axis using dilated causal convolutions, which enable the model to capture long-range dependencies without increasing depth significantly.

Each TCN layer applies a 1D convolution with a kernel size k and dilation rate r, preserving the temporal order of observations. Multiple layers are stacked with increasing dilation rates, allowing the network to aggregate information from a wide temporal context.

The TCN module outputs a temporally enriched representation $\mathbf{Z} \in \mathbb{R}^{N \times r}$ for the current time step, which is then passed to the prediction heads.

The key advantages of using TCN in this setting include:

- Efficient parallelism in training and inference;
- Ability to capture both short- and long-term temporal patterns;
- Stable gradients due to convolutional structure.

Figure 4.4 illustrates the temporal modelling process using stacked TCN layers applied to spatially encoded features.

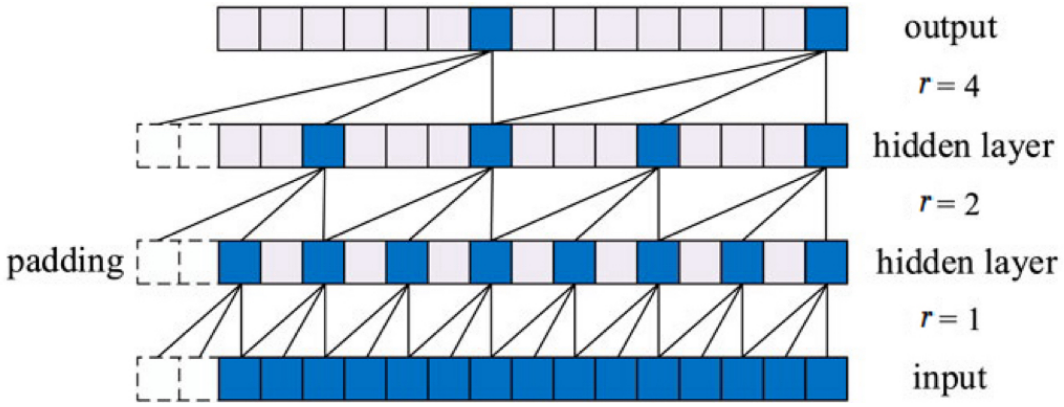

Figure 4.4: Architecture of the Temporal Convolutional Network (TCN) with stacked dilated convolutional layers. The network processes input sequences through multiple hidden layers with exponentially increasing dilation rates ($r = 1, 2, 4$), enabling the capture of multi-scale temporal dependencies while maintaining a compact architecture. Each layer's dilation rate doubles, exponentially expanding the receptive field to capture both short-term and long-range patterns in the sensor data.

4.5 Prediction Heads and Training Objective

After encoding spatial and temporal dependencies through the GCN and TCN modules, the MLAD framework introduces a dual-head prediction mechanism designed to strengthen the model's robustness and sensitivity to heterogeneous anomalies. This

module serves as a bridge between the spatial-temporal encoder and the anomaly scoring stage, transforming latent representations into concrete predictive tasks that encourage the network to learn stable yet discriminative temporal features. In multivariate industrial systems such as SWaT and WADI, anomalies manifest in various forms, some are abrupt and transient, while others evolve gradually over time. To capture this diversity, MLAD employs two complementary prediction heads that share the same latent representation $Z \in \mathbb{R}^{N \times d}$, where N is the number of sensors and d denotes the latent feature dimension.

The first branch, referred to as the reconstruction head, aims to recover the input sequence within each sliding window. By reconstructing the observed historical data, it forces the encoder to preserve temporal stability and smooth continuity, which is particularly effective for identifying contextual or low-amplitude anomalies that develop slowly. In contrast, the forecasting head focuses on predicting sensor readings at the next time step (or over several future steps). This branch emphasises the modelling of short-term temporal transitions, allowing the model to capture sudden deviations that violate expected local patterns. By operating in parallel, these two heads promote complementary feature learning: reconstruction encourages the preservation of global consistency, whereas forecasting amplifies responsiveness to local changes. This dual-task mechanism effectively regularises the representation learning process, ensuring that the network does not overfit to a single anomaly type but instead learns balanced temporal dynamics that generalise across multiple behavioural regimes.

Both prediction heads are connected to the same shared spatial-temporal encoder but maintain independent output layers. During training, their learning signals are integrated through a multi-task objective that combines reconstruction and forecasting losses into a unified optimisation target. While the exact mathematical formulation and optimisation details are discussed in Section 5.3, it is important to emphasise that the simultaneous training of these two tasks encourages the encoder to learn features that are both stable and sensitive to change. This architectural design provides MLAD with the ability to capture multi-scale temporal dependencies while preserving interpretability, as each prediction head contributes distinct and complementary insights into the underlying system behaviour.

4.6 Training Strategy and Optimisation

To effectively train the MLAD model and ensure convergence of both reconstruction and forecasting tasks, a two-stage training strategy is employed.

Stage 1: Pre-training on Normal Data. Since anomaly detection typically involves highly imbalanced datasets where normal samples dominate, MLAD is first trained on a subset of purely normal data. This stage allows the model to learn a robust latent representation of normal patterns without being affected by noisy anomaly labels. During this stage, both prediction heads are optimised using the multi-task loss $\mathscr{L}_{\text{total}}$.

Stage 2: Fine-tuning with Full Dataset. After pre-training, the model is fine-tuned using the full dataset, which includes both normal and anomalous segments. At this stage, learning rate scheduling and early stopping mechanisms are used to avoid overfitting to outlier patterns. The same multi-task loss is applied, but the model now gains exposure to the statistical diversity present in real-world sensor streams.

Training Details. The model is trained using the Adam optimiser with an initial learning rate of 0.001. A mini-batch size of 64 is adopted, and training is performed for up to 200 epochs. A cosine learning rate decay schedule is used to gradually reduce the learning rate and stabilise convergence. Dropout is applied in both GCN and TCN modules to improve generalisation.

This training process enables the MLAD model to gradually acquire spatial, temporal, and predictive understanding of normal system behaviour, resulting in higher anomaly sensitivity and lower false positive rates during evaluation.

4.7 Summary

The MLAD framework combines clustering-based graph construction, spatial-temporal modelling, and multi-task prediction to address the challenges of multivariate time series anomaly detection. By leveraging unsupervised clustering, the model identifies behaviourally similar sensor groups and constructs cluster-constrained graphs that preserve local semantic structure. GCN layers are then applied to encode intra-cluster spatial dependencies, while TCN layers capture long-range temporal patterns from sequential data.

The dual prediction heads reconstruction and forecasting serve complementary purposes: the former promotes input consistency and stability, while the latter anticipates behavioural transitions. Their joint optimisation via multi-task learning strengthens the feature representation and amplifies anomaly-related discrepancies. The two-stage training strategy, beginning with pre-training on normal data and followed by fine-tuning on full datasets, ensures robust convergence and generalisation.

Through this integrated design, MLAD effectively bridges the gap between graph-based representation learning and time-series forecasting, delivering improved accuracy and interpretability in detecting diverse types of anomalies in cyber-physical systems.

CHAPTER 5

MULTI-TASK FORECASTING

In multivariate time-series anomaly detection, a fundamental challenge lies in learning robust representations from limited normal samples that can sensitively respond to various types of anomalies. Traditional approaches typically rely on a single objective, such as forecasting error or reconstruction error, to generate anomaly scores. However, these single-task strategies often show clear limitations when confronted with heterogeneous anomalies in complex systems.

To overcome these limitations, the proposed MLAD (Multi-task Learning for Anomaly Detection) model introduces a **Multi-task Learning (MTL)** strategy, built upon a dual-path structure involving both forecasting and reconstruction tasks. The central idea is to guide the model to simultaneously optimise temporal forecasting and historical reconstruction objectives, thereby encouraging it to learn both stable and sensitive feature representations. Compared to single-objective frameworks, this design enables the model to extract complementary insights from different anomaly perspectives, improving its generalisation to both abrupt and gradual anomalies.

This chapter presents a detailed analysis of the multi-task learning module in MLAD. Section 5.1 first discusses the motivation behind adopting a multi-task setup, including insights from the WADI and SWaT datasets where anomalies exhibit diverse temporal patterns. Section 5.2 then describes the dual-head output architecture and how the forecasting and reconstruction branches are structurally coordinated. In Section 5.3, I introduce the joint loss function and provide its mathematical formulation.

Following the design discussion, Section 5.4 presents quantitative evaluation re-

sults, demonstrating the effectiveness of multi-task learning in boosting overall model performance. Section 5.5 offers visual case studies that highlight the complementary strengths of each prediction head under various anomaly scenarios. In Section 5.6, I compare MLAD to existing forecasting-based anomaly detection models, illustrating its advantage in complex datasets in terms of both accuracy and interpretability.

This chapter systematically explores how the multi-task learning strategy contributes to the MLAD model. Through theoretical explanation, quantitative analysis, and empirical evidence, I show how this mechanism improves anomaly sensitivity and robustness, serving as a foundational component for subsequent anomaly scoring and detection procedures.

5.1 Motivation for Multi-task Forecasting

In real-world multivariate sensor systems, anomaly patterns vary significantly across different sensors and scenarios. Some anomalies appear as sharp spikes that are immediately detectable by forecasting errors, while others manifest as subtle drifts or sustained deviations that are more effectively captured by reconstruction errors. Relying solely on either forecasting or reconstruction leads to blind spots where specific anomaly types may be missed.

To illustrate this complementarity, I conduct an analysis using two representative cases from multivariate time-series datasets:

- **Case A:** A sudden actuator override causes a sharp increase in the sensor reading. Forecasting-based methods focused on next-step prediction respond immediately to the deviation.

- **Case B:** A contextual anomaly develops gradually over time (e.g., leakage or drift). While forecasting models fail to detect it due to minimal short-term change, reconstruction-based methods accumulate errors and flag the anomaly effectively.

These two cases highlight how forecasting and reconstruction tasks capture different aspects of abnormality. Forecasting excels at detecting temporal discontinuities, while reconstruction identifies deviations from consistent patterns over longer periods. Therefore, combining both tasks enables the model to achieve higher coverage and sensitivity to heterogeneous anomalies.

Moreover, this dual-task design facilitates robust representation learning: forecasting drives the network to capture short-term temporal evolution, while reconstruction encourages memorisation of stable historical trends. Together, they regularise the learning process and improve generalisation.

Figure 5.1 provides a visual comparison between these two tasks applied to multivariate sensor data.

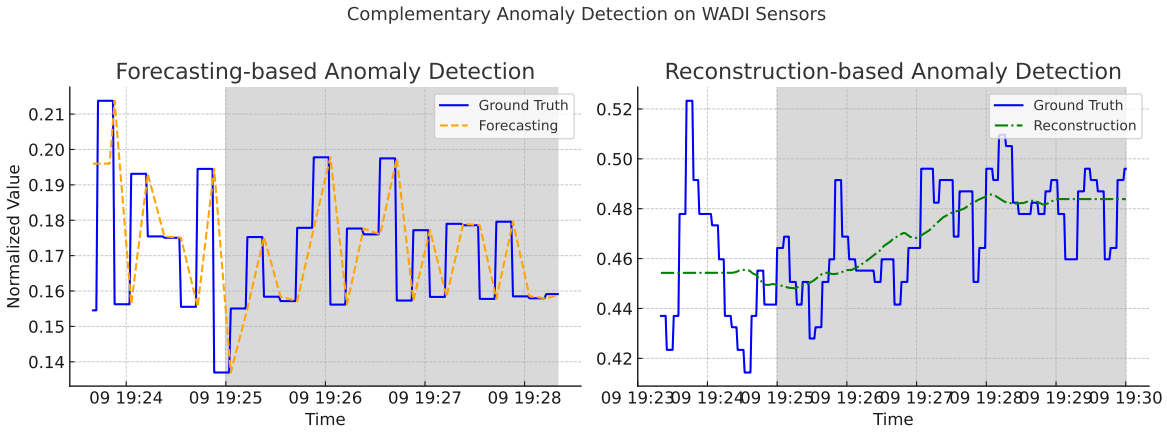

Figure 5.1: Illustration of anomaly detection using forecasting (left) and reconstruction (right). The forecasting-based method captures abrupt spikes and collective anomalies, while the reconstruction-based method effectively detects contextual anomalies over extended periods. This motivates the use of both tasks in MLAD.

In summary, multi-task learning provides a principled approach to improving anomaly detection sensitivity and generalisation. By leveraging the strengths of both forecasting and reconstruction, MLAD is better positioned to handle the complex behaviours observed in real-world cyber-physical systems.

5.2 Architecture Design of Multi-task Heads

The MLAD model integrates multi-task learning by employing two parallel prediction heads: a forecasting head and a reconstruction head. These heads are designed to operate on the same spatial-temporal feature representations but are optimised for distinct objectives. This architectural design encourages the model to capture complementary aspects of time-series behaviour, thus improving its robustness and sensitivity to diverse anomaly types.

The multi-task architecture is built upon a shared encoder backbone, which includes the spatial graph encoder (GCN layers) and the temporal sequence encoder (TCN layers), as described in Chapter4. This shared module processes the input sequence and generates a latent representation $\mathbf{Z} \in \mathbb{R}^{N \times d}$, where N is the number of sensors and d is the dimension of the latent space. The shared encoder ensures that both tasks forecasting and reconstruction are grounded in a common understanding of sensor relationships and temporal patterns, thereby aligning their learning directions.

The forecasting head takes the latent representation $\mathbf{Z}$ and predicts the sensor values at the next time step $t+1$, formulated as:

$$\hat{\mathbf{x}}_{t+1} = \mathbf{W}_{\text{pred}}\mathbf{Z} + \mathbf{b}_{\text{pred}}, \tag{5.1}$$

where $\mathbf{W}_{\text{pred}} \in \mathbb{R}^{d \times 1}$ and $\mathbf{b}_{\text{pred}}$ are learnable parameters. This design enables the model to focus on transition dynamics and sharp variations in sensor behaviour.

In contrast, the reconstruction head aims to recreate the input sequence over the window $[t-T+1, ..., t]$ from the same latent representation $\mathbf{Z}$, using:

$$\hat{\mathbf{X}}_{t-T+1\,t} = \text{MLP}_{\text{recon}}(\mathbf{Z}), \tag{5.2}$$

where $\text{MLP}_{\text{recon}}$ is a multi-layer perception that expands the latent vector into the reconstructed time-series window. This task emphasises the preservation of stable patterns and long-term consistency in sensor behaviour.

The dual-head design brings several advantages. First, it offers **complementary supervision**: the forecasting head is sensitive to temporal evolution, while the reconstruction head captures static consistency, together covering a broader spectrum of anomaly types. Second, **parameter efficiency** is achieved by sharing the encoder, which reduces model complexity and improves convergence. Third, the structure provides **robust learning signals** by allowing each task to guide the other, stabilising training and reducing overfitting.

In summary, the dual-task prediction architecture of MLAD leverages a shared spatial temporal encoder to generate representations that are jointly used for short-term forecasting and input reconstruction. This design enables the model to better capture both sudden shifts and gradual drifts in sensor dynamics, which are often missed by single-task models, ultimately enhancing the anomaly detection performance in complex multivariate time-series environments.

5.3 Loss Function and Training Objective

Building on the architectural foundation introduced in Chapter 4, this section details the mathematical formulation and optimisation strategy that jointly train the reconstruction and forecasting heads within the MLAD framework. The effectiveness of MLAD relies on the careful integration of its multi-task objectives: forecasting and reconstruction. To enable joint optimisation of these tasks, a multi-task loss function is designed that balances their respective contributions. This section outlines the mathematical formulation of the loss components, the rationale behind their weighting, and the optimisation strategies employed to train the model efficiently and robustly.

The model outputs two sets of predictions from the latent representation $\mathbf{Z}$: the reconstructed input sequence $\hat{\mathbf{X}}_{t-T+1:t}$ and the forecasted next time step $\hat{\mathbf{x}}_{t+1}$. These are compared against the corresponding ground truth values using Mean Squared Error (MSE) as the loss metric for both tasks.

Reconstruction Loss

The reconstruction task aims to recover the input sequence from the latent space. The reconstruction loss $\mathscr{L}\text{rec}$ is defined as:

$$\mathscr{L}_{\text{rec}} = \frac{1}{T} \sum_{i=t-T+1}^{t} |\hat{\mathbf{x}}_i - \mathbf{x}_i|^2, \tag{5.3}$$

where $\hat{\mathbf{x}}_i$ denotes the reconstructed sensor vector at time i, and $\mathbf{x}_i$ is the ground truth.

Forecasting Loss

For forecasting, the model predicts the next time step $\hat{\mathbf{x}}_{t+1}$ based on the temporal representation $\mathbf{Z}$. The forecasting loss $\mathscr{L}\text{pred}$ is given by:

$$\mathscr{L}_{\text{pred}} = |\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}|^2. \tag{5.4}$$

Total Loss

To jointly optimise the two objectives, a weighted sum of the reconstruction and forecasting losses is used. The total loss $\mathscr{L}\text{total}$ is defined as:

$$\mathscr{L}_{\text{total}} = \lambda \mathscr{L}_{\text{rec}} + (1-\lambda)\mathscr{L}_{\text{pred}}, \tag{5.5}$$

where $\lambda \in [0,1]$ is a tunable hyper-parameter that balances the contribution of each task. In this study, $\lambda = 0.3$ is chosen to give equal weight to both losses, which empirically led to more stable and effective training across datasets.

Optimisation Strategy

The MLAD model is trained using the Adam optimiser with an initial learning rate of 0.0005. The training proceeds in two phases:

- **Pre-training:** The model is first trained on a clean subset of normal data to capture stable patterns and avoid bias from anomalies.
- **Fine-tuning:** It is then fine-tuned on the full dataset, including potential anomalies, using early stopping and learning rate scheduling to prevent overfitting.

A cosine annealing learning rate scheduler is applied to gradually reduce the learning rate over time. Batch size is set to 64, and training is conducted for up to 20 epochs. Dropout is applied within the encoder modules to regularise learning and improve generalisation.

This loss and optimisation design ensures that the model learns temporal and structural patterns simultaneously, thereby enhancing its ability to detect both abrupt and gradual anomalies in multivariate time-series data.

5.4 Quantitative Evaluation and Anomaly Scoring

To evaluate the effectiveness of the proposed dual-task learning strategy, this section presents a comprehensive quantitative analysis of forecasting and reconstruction performance, along with the derived anomaly scores on benchmark datasets. The comparison is made between three model configurations: using only forecasting, using only reconstruction, and using both in a unified multi-task framework (MLAD). Additionally, this section explains how the prediction errors are transformed into anomaly scores, and how detection performance is assessed using standard metrics.

The anomaly score in MLAD is computed by combining the forecasting and reconstruction errors, with each task contributing a distinct perspective. The forecasting head aims to predict the next time step value $\hat{\mathbf{x}}_{t+1}$, while the reconstruction head attempts to recover the input sequence $\hat{\mathbf{x}}_{t-T+1:t}$. Let L_{pred} and L_{rec} denote the forecasting and

reconstruction error terms used to form the anomaly score at time step t. The anomaly score $\mathscr{S}_t$ at time t is defined as:

$$\mathscr{S}_t = \alpha \cdot \mathscr{L}_{\text{rec}} + (1-\alpha) \cdot \mathscr{L}_{\text{pred}} \tag{5.6}$$

where α is a weight parameter (empirically set to 0.3) to balance both contributions. It is important to distinguish between the window-level training loss and the time-step anomaly score used during inference. In Section 5.3, L_{rec} is defined as the average reconstruction loss over the input window, which is used to stabilise optimisation during training. However, for anomaly scoring at time step t, the reconstruction contribution is computed from the instantaneous reconstruction residual at the current timestamp, rather than from the full window average. Therefore, in Eq. (5.6), L_{rec} should be interpreted as the reconstruction error associated with time step t, while the window-averaged form in Eq. (5.3) is used only for parameter learning. This distinction ensures that $\mathscr{S}_t$ remains a fine-grained per-time-step anomaly score, while the training objective still benefits from the stability of window-level averaging. This scoring strategy ensures that the model can detect both abrupt anomalies (better captured by forecasting) and gradual drifts or pattern changes (highlighted by reconstruction errors).

To determine whether a given time step is anomalous, the anomaly scores are compared against a detection threshold. In my evaluation, this threshold is determined using the Peak Over Threshold (POT) method, which adapts to score distribution in an unsupervised manner. An observation $\mathbf{x}_t$ is marked as anomalous if its score $\mathscr{S}_t$ exceeds this threshold.

The detection performance is measured using standard metrics including Area Under the Receiver Operating Characteristic Curve (ROC-AUC) and Area Under the Precision-Recall Curve (PRC-AUC). These metrics are robust to class imbalance and suitable for evaluating anomaly detectors, where true anomalies are sparse relative to normal samples.

I conduct an ablation study to compare the performance of different task settings. Table 5.1 reports the detection scores of forecasting-only, reconstruction-only, and multi-task (MLAD) models on the SWaT and WADI datasets. Each result is averaged over three runs with different seeds.

Table 5.1: Ablation study: AUC performance of different task configurations. Best results in each column are **bolded**.

Model	**SWaT**		**WADI**	
	ROC-AUC	PRC-AUC	ROC-AUC	PRC-AUC
Forecasting Only	0.8124	0.7125	0.4273	0.0521
Reconstruction Only	0.8409	0.6970	0.6388	0.3775
MLAD (Forecast + Recon)	**0.8583**	**0.7570**	**0.8327**	**0.5267**

The results clearly demonstrate that the multi-task model outperforms both single-task variants across all metrics. In particular, the improvements in PRC-AUC are substantial, indicating that MLAD is more effective at distinguishing true anomalies from false positives, especially under class imbalance conditions.

In addition to numerical metrics, I visualise the prediction and reconstruction errors over time to show how different types of anomalies are captured. Figure5.1 illustrates two sample cases: one where forecasting detects sudden spikes missed by reconstruction, and another where reconstruction better captures gradual drifts that forecasting overlooks. These case studies validate the complementary strengths of the two tasks.

In summary, the quantitative evaluation highlights the effectiveness of the dual-task architecture in enhancing anomaly sensitivity. The integrated scoring mechanism enables the model to respond to a wider range of anomaly types. Together with robust threshold and standardised metrics, this approach yields accurate and reliable detection results in complex cyber-physical systems such as SWaT and WADI.

5.5 Case Studies on Forecasting and Reconstruction

To further illustrate the complementary nature of forecasting and reconstruction tasks in the MLAD model, I present several case studies drawn from the SWaT and WADI datasets. These examples demonstrate how each task contributes uniquely to the detection of different anomaly types, thus validating the need for a dual-task architecture.

In the first case, I examine a point anomaly caused by a sudden actuator malfunction. As shown in Figure 5.2, the forecasting head successfully predicts a sharp deviation in sensor readings, resulting in a high anomaly score. However, the reconstruction head fails to flag this anomaly effectively due to its emphasis on smoothing and stability. This case underscores the forecasting head's strength in capturing short-term irregularities.

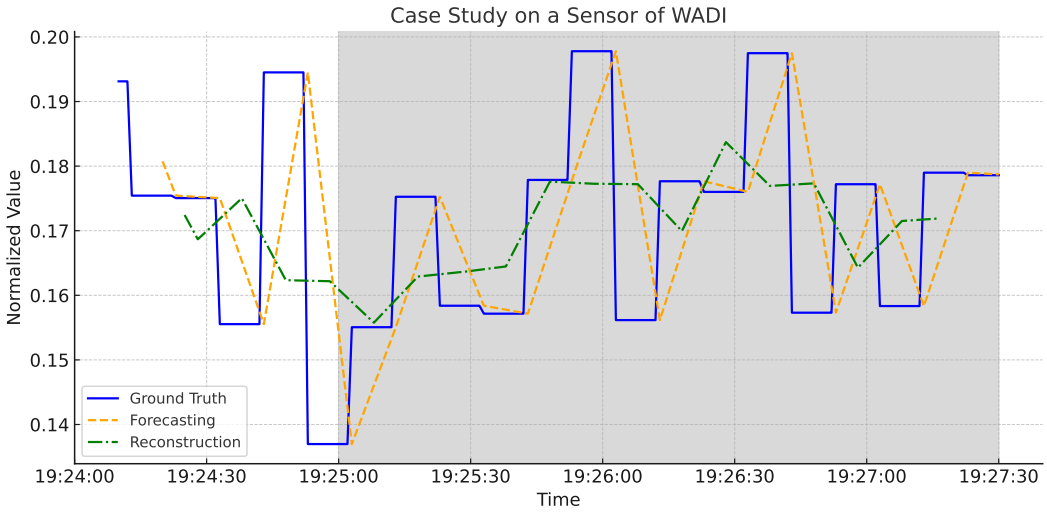

Figure 5.2: Case 1: Case study on a WADI sensor showing a spike anomaly. The forecasting head captures the sudden deviation, while the reconstruction head fails to respond to the abrupt change. The shaded region indicates the anomalous period.

In contrast to the spike anomaly illustrated in Figure 5.2, Figure 5.3 depicts a slow drift anomaly caused by calibration error in a chemical concentration sensor. While the forecasting head, which emphasises short-term dynamics, underestimates the anomaly due to the gradual nature of the drift, the reconstruction head progressively diverges from the ground truth as the drift persists. This behaviour underlines the complementary advantage of reconstruction in identifying low-amplitude but long-lasting anomalies.

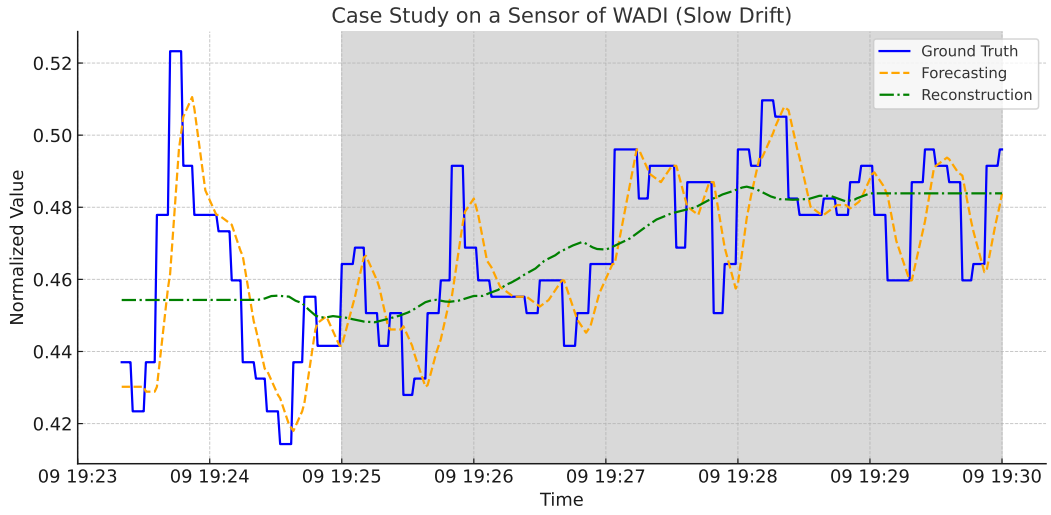

Figure 5.3: Case 2: A sensor showing a slow drift anomaly. The reconstruction head accumulates greater deviation from the ground truth than the forecasting head, highlighting its effectiveness in capturing persistent changes. The shaded region indicates the anomalous period.

These cases highlight an important insight: forecasting excels in identifying abrupt, transient faults, while reconstruction is better suited for subtle and sustained deviations. Relying solely on either task would miss critical signals present in industrial time-series data. The dual-task configuration of MLAD thus ensures a broader detection spectrum and stronger anomaly sensitivity.

In real-world deployments, such as water treatment plants or power systems, faults manifest in diverse forms. Some anomalies occur as immediate sensor spikes; others develop progressively over minutes or hours. By incorporating both forecasting and reconstruction heads, MLAD is capable of capturing this range of behaviours, ensuring not only early warning but also interpretability through task-specific error analysis.

Therefore, the integration of both outputs not only improves detection accuracy but also provides more actionable insights for system operators. By analysing the source of error(forecasting and reconstruction) operators can infer the type of fault and respond accordingly. This interpretability is particularly valuable in mission-critical systems where understanding the cause of anomalies can guide remediation efforts.

5.6 Comparison with Existing Forecasting-based Models

To further contextualise the performance and design choices of MLAD, this section compares its forecasting strategy against several existing models that rely primarily or exclusively on forecasting-based anomaly detection. I particularly focus on GDN [18] and MTAD-GAT [20], both of which have been influential in graph-based time series forecasting for anomaly detection.

GDN formulates the anomaly detection problem as a forecasting task by learning a node-level graph and predicting future values. Its prediction error is used to infer anomalies. While GDN incorporates a graph structure and self-attention mechanisms to model spatial correlations, it lacks a reconstruction branch. This design makes it sensitive to point-wise deviations but less robust when facing gradual or distributional anomalies, which often require a longer historical context for identification.

Similarly, MTAD-GAT introduces a dual attention mechanism temporal and spatial to enhance its forecasting accuracy. It models multi-hop spatial relationships via graph attention and temporal dependencies via sequence encoders. However, like GDN, MTAD-GAT optimises only a forecasting objective. In my experiments on the SWaT and WADI datasets, I observed that while MTAD-GAT performs well in capturing local temporal inconsistencies, it struggles with slow drifts or sustained behavioural shifts that do not produce sharp prediction errors.

MLAD distinguishes itself by adopting a multi-task architecture that includes both forecasting and reconstruction branches. This design enables it to detect a broader range of anomalies. The forecasting task captures short-term deviations, while the reconstruction task models longer-term temporal regularity. This synergy between tasks improves robustness, reduces the risk of overfitting to a single anomaly type, and enhances interpretability by isolating the source of deviation.

To support these observations, Table 5.2 presents a comprehensive comparison of AUC-ROC and AUC-PRC scores across three benchmark datasets: SWaT, WADI, and SMD. The table highlights the average performance and standard deviation across five independent runs for each method. Models such as PCA, AutoEncoder, and USAD serve as classical or baseline methods, while GDN, MTAD-GAT, and CST-GL represent more recent graph- and forecasting-based approaches. MLAD achieves the highest AUC-ROC across all three datasets and top or near-top AUC-PRC values, showcasing its superior and balanced detection capability.

Table 5.2: Average AUC performance (± standard deviation) of five experimental runs on three benchmark datasets. The best, second best, and third best results in each column are marked in **bold**, with and † respectively.

	SWaT		**WADI**		**SMD**	
	AUC-ROC	AUC-PRC	AUC-ROC	AUC-PRC	AUC-ROC	AUC-PRC
PCA	0.8257±0.0000	0.7298±0.0000	0.5597±0.0000	†0.2731±0.0000	0.6742±0.0000	0.2189±0.0000
Kmeans	0.7391±0.0000	0.2418±0.0000	†0.6030±0.0000	0.1158±0.0000	0.5855±0.0000	0.1308±0.0000
AutoEncoder	0.8311±0.0088	0.7224±0.0094	0.5291±0.0285	0.2210±0.0205	0.8270±0.0008	0.4388±0.0046
USAD	0.8213±0.0056	0.7087±0.0055	0.5535±0.0103	0.1945±0.0008	0.7888±0.0077	0.4686±0.0011
MTAD-GAT	0.8261±0.0040	0.7176±0.0043	0.4119±0.0295	0.0729±0.0013	†0.8576±0.0035	†0.5057±0.0082
THOC	†0.8380±0.0051	†0.7440±0.0063	0.4840±0.0112	0.1440±0.0020	0.8512±0.0045	0.4852±0.0063
GDN	0.8124±0.0177	0.7135±0.0035	0.4725±0.0056	0.0521±0.0070	0.8443±0.0150	0.4684±0.0142
CST-GL [63]	0.8520±0.0022	**0.7628**±0.0032	0.8283±0.0179	**0.5477**±0.0197	0.8604±0.0131	0.5132±0.0273
MLAD	**0.8583**±0.0025	0.7570±0.0050	**0.8327**±0.0040	*0.5267±0.0034	**0.8703**±0.0078	**0.5204**±0.0031

The tabulated results further validate MLAD's generalisation and effectiveness across datasets of varying complexity and scale. In particular, MLAD's consistent performance on both SWaT and WADI two structurally distinct CPS datasets demonstrates its adaptability. Unlike some models that excel only in either low-noise (SWaT) or high-complexity (WADI) settings, MLAD balances both with minimal performance degradation.

In conclusion, while forecasting-based models such as GDN and MTAD-GAT offer efficient and scalable solutions, their single-objective design limits their sensitivity to diverse anomaly types. MLAD overcomes this limitation by unifying forecasting and reconstruction, achieving better coverage, lower false positives, and enhanced interpretability. This dual-path strategy reflects a more nuanced understanding of multivariate time series behaviour, particularly in complex industrial environments such as those represented by SWaT and WADI.

5.7 Summary

In this chapter, I introduced and analysed the multi-task forecasting module within the proposed MLAD framework. Built upon the spatial modelling foundation discussed in Chapter 4, this module enhances the model's ability to capture both temporal trends and structural shifts through the dual tasks of sequence forecasting and input reconstruction.

I began with an overview of the motivation behind multi-task learning in anomaly detection. Traditional single-task architectures often fail to detect the full spectrum of anomalies, particularly when those anomalies exhibit diverse characteristics such as sudden spikes or gradual drifts. Forecasting is typically more responsive to abrupt temporal deviations, while reconstruction excels at maintaining consistency in normal

behavioural patterns. By combining these two objectives, the MLAD model creates a richer and more robust representation space.

The architecture of the dual-task heads was then presented. Sharing a common encoder, the model branches into two separate decoders: one for forecasting future time steps and the other for reconstructing the input sequence. This architecture allows each task to reinforce the other's learning, improving both stability and generalisation. I also replaced visual representation of the design with descriptive text to emphasise clarity and avoid unnecessary visual complexity.

I further defined the joint loss function that balances the two learning objectives using a weighted sum. This formulation supports flexibility in tuning the model's sensitivity to different types of anomalies. In practice, an equal weighting scheme ($\lambda = 0.5$) yielded strong results across multiple benchmarks.

In the subsequent evaluation, I demonstrated the advantages of the dual-task setup over single-task variants using both ablation studies and ROC/PRC comparisons. Case studies highlighted how each task captures unique anomaly characteristics, with one detecting deviations missed by the other. Together, they provide complementary perspectives that improve interpretability and diagnostic effectiveness.

Moreover, I embedded the scoring and anomaly detection strategy directly into this chapter to present a unified view of the full forecasting pipeline. The anomaly score is derived by combining errors from both reconstruction and forecasting, enhancing sensitivity to both point anomalies and gradual deviations. A simple yet effective threshold mechanism is then applied to these scores for final anomaly labelling.

In summary, the multi-task forecasting module plays a central role in the MLAD framework, bridging spatial representations with temporal inference and anomaly scoring. Its ability to learn diverse patterns and provide dual perspectives on abnormality makes it a key contributor to the model's superior performance on benchmarks such as SWaT and WADI. In the next chapter, I conclude the thesis by summarising my contributions, discussing limitations, and outlining potential future research directions.

CHAPTER 6

CONCLUSION

This thesis has tackled the significant challenge of detecting anomalies in multivariate time-series data. Motivated by the shortcomings of existing methods,which often assume sensor homogeneity or rely on simplistic, predefined graph structures. I introduced a novel and comprehensive framework named MLAD (Multi-task Learning for Anomaly Detection). The core innovation of this work lies in a cohesive pipeline that seamlessly integrates unsupervised sensor behaviour clustering with cluster-constrained graph neural networks and multi-task learning. This integrative approach allows MLAD to effectively model the complex spatial-temporal dependencies inherent in real-world sensor data, leading to robust, accurate, and interpretable anomaly detection.

The research began by establishing that sensors within a system often exhibit distinct behavioural patterns. Rather than forcing all sensors into a single model, I developed an unsupervised method to group them based on their inherent dynamics. This process involved extracting informative statistical features from sliding time windows, reducing their dimensionality using UMAP to preserve local relationships, and finally applying DBSCAN to discover natural sensor clusters without requiring prior knowledge of the system‚Äôs structure. This foundational step proved crucial, not only enhancing the interpretability of the model by aligning data-driven clusters with physical subsystems but also providing the structural blueprint for the subsequent graph-based learning stage.

Building upon these behavioural clusters, I constructed a graph where connections are permitted only between sensors within the same cluster. This design choice creates a

sparse, semantically meaningful graph structure that confines information propagation to functionally related sensors, preventing the model from being diluted by noise from unrelated parts of the system. I then employed a Graph Convolutional Network (GCN) to encode these spatial dependencies and a Temporal Convolutional Network (TCN) to capture evolving temporal patterns. This spatial-temporal graph model forms the expressive core of MLAD, capable of learning rich representations that are sensitive to both the local context within a sensor group and its evolution over time.

To ensure sensitivity to a broad spectrum of anomalies from sudden spikes to gradual drifts. I designed a multi-task learning objective in which the model is trained simultaneously to forecast future sensor values and reconstruct past input sequences. These two tasks provide complementary signals: forecasting excels at identifying abrupt temporal deviations, while reconstruction is adept at detecting subtle, sustained changes in patterns. The anomalies are subsequently detected by a combined score derived from the errors of both tasks. Extensive evaluation on the SWaT and WADI datasets demonstrated that this multi-task strategy, underpinned by the cluster-informed graph structure, consistently outperforms strong baselines and state-of-the-art methods.

Despite these encouraging results, the MLAD framework presents several opportunities for future enhancement. The current clustering process operates offline, meaning that sensor groupings are determined prior to training. Although this static configuration effectively captures behavioural modularity, it does not account for temporal evolution in sensor relationships. Future work could focus on developing an online or adaptive clustering mechanism that dynamically updates sensor associations as system conditions change, enabling the model to better adapt to non-stationary environments.

Additionally, the model's performance depends on several hyperparameters including UMAP dimensionality, DBSCAN thresholds, and task weighting coefficients, which currently require manual tuning. Future research could explore automated hyperparameter optimisation or meta-learning strategies to enable MLAD to self-adjust these parameters across different datasets and system scales. Such adaptive optimisation would improve robustness and ease of deployment, especially in industrial contexts where expert intervention is limited.

Finally, although MLAD has been validated on two benchmark industrial control datasets (SWaT and WADI), its underlying design principles behaviour based clustering, cluster-constrained graph learning, and multi-task forecasting are not domain-specific. Future studies should investigate the cross-domain applicability of MLAD to other complex sensor networks, such as energy systems, healthcare monitoring, and IT infras-

tructure. These domains share similar challenges of high dimensionality, heterogeneity, and temporal dependency, making them suitable candidates for validating MLAD's generalisation capacity and interpretability.

In summary, future research on MLAD should aim to enhance its adaptability, automation, and cross-domain generalisation. By building on the strong foundation established in this thesis, such extensions would further improve the framework's resilience and practicality, supporting reliable, interpretable, and data-driven anomaly detection in increasingly complex cyber-physical environments.

BIBLIOGRAPHY

[1] C. C. AGGARWAL, *Outlier Analysis*, Springer, 2017.

[2] C. C. AGGARWAL, A. HINNEBURG, AND D. A. KEIM, *On the surprising behavior of distance metrics in high dimensional space*, Proceedings of the 8th International Conference on Database Theory (ICDT), (2001), pp. 420–434.

[3] C. M. AHMED, V. R. PALLETI, AND A. P. MATHUR, *Wadi: A water distribution testbed for research in the design of secure cyber physical systems*, in Proc. 3rd Int. Workshop Cyber-Phys. Syst. Smart Water Networks (CySWATER), 2017, pp. 25–28.

[4] L. AKOGLU, H. TONG, AND D. KOUTRA, *Graph-based anomaly detection and description: A survey*, Data Mining and Knowledge Discovery, 29 (2015), pp. 626–688.

[5] J. AN AND S. CHO, *Variational autoencoder based anomaly detection using reconstruction probability*, in Special Lecture on IE, vol. 2, 2015, pp. 1–18.

[6] F. R. BACH AND M. I. JORDAN, *Learning graphical models for stationary time series*, IEEE Trans. Signal Process., 52 (2004), pp. 2189–2199.

[7] K. BEYER, J. GOLDSTEIN, R. RAMAKRISHNAN, AND U. SHAFT, *When is nearest neighbor meaningful?*, in Lecture notes in computer science, Springer, 1999, pp. 217–235.

[8] A. BLAZQUEZ GARCIA, A. CONDE, U. MORI, AND J. A. LOZANO, *A review on outlier/anomaly detection in time series data*, ACM Computing Surv., 54 (2021), pp. 1–33.

[9] R. J. BOLTON AND D. J. HAND, *Statistical fraud detection: A review*, Statistical Science, 17 (2002), pp. 235–255.

[10] M. M. BREUNIG, H.-P. KRIEGEL, R. T. NG, AND J. SANDER, *Lof: Identifying density-based local outliers*, in Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.

[11] M. M. BREUNIG, H.-P. KRIEGEL, R. T. NG, AND J. SANDER, *Lof: Identifying density-based local outliers*, in Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.

[12] R. CHALAPATHY AND S. CHAWLA, *Deep learning for anomaly detection: A survey*, arXiv preprint arXiv:1901.03407, (2019).

[13] V. CHANDOLA, A. BANERJEE, AND V. KUMAR, *Anomaly detection: A survey*, ACM Comput. Surv., 41 (2009), pp. 1–58.

[14] Y. CHEN, W. WANG, AND S. HAN, *Graph learning: A survey*, arXiv preprint arXiv:2007.03098, (2020).

[15] Z. CHEN, X. YANG, D. CAO, C. SHEN, Y. YU, AND Z. ZHENG, *Learning graph structures with transformer for multivariate time series anomaly detection in iot*, in Proc. 2021 IEEE Int. Conf. Data Mining (ICDM), 2021, pp. 173–182.

[16] E. CHOI, M. T. BAHADORI, J. SUN, J. KULAS, A. SCHUETZ, AND W. STEWART, *Doctor ai: Predicting clinical events via recurrent neural networks*, in Proc. 1st Machine Learning for Healthcare Conference (MLHC), 2016, pp. 301–318.

[17] Z. Z. DARBAN, M. GHASEMI, A. NAZARI, AND R. AKBARI, *Deep learning for time series anomaly detection: A survey*, ACM Comput. Surv., 57 (2024), p. Article 15.

[18] A. DENG AND B. HOOI, *Graph neural network-based anomaly detection in multivariate time series*, in Proc. AAAI Conf. Artif. Intell., vol. 35, 2021, pp. 4027–4035.

[19] R. DEY AND F. M. SALEM, *Gate-variants of gated recurrent unit (gru) neural networks*, in 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), IEEE, 2017, pp. 1597–1600.

[20] C. DING, S. SUN, AND J. ZHAO, *Mst-gat: A multimodal spatial-temporal graph attention network for time series anomaly detection*, arXiv preprint arXiv:2310.11169, (2023).

[21] S. EL-SAPPAGH, T. ABUHMED, S. RIAZUL ISLAM, AND K. S. KWAK, *Multimodal multitask deep learning model for alzheimer's disease progression detection based on time series data*, Neurocomputing (Amsterdam), 412 (2020), pp. 197–215.

[22] E. ESKIN, A. ARNOLD, M. PRERAU, L. PORTNOY, AND S. J. STOLFO, *Anomaly detection over noisy data using learned probability distributions*, in Proceedings of the International Conference on Machine Learning (ICML), 2000.

[23] C. GAO, X. HE, Y. ZHANG, K. WANG, X. WANG, AND M. WANG, *A survey of graph neural networks for recommender systems: Challenges, methods, and directions*, ACM Trans. Recomm. Syst., 1 (2023), pp. 1–51.

[24] N. . GITHUB, *Server machine dataset (smd) for multivariate time series anomaly detection*.
GitHub repository, 2019.
Datasets: SMAP, MSL, SMD ‚Äì 28 machines, 38 dimensions each, anomaly ratio ‚âà 4.16%.

[25] H. GUO, C. YANG, F. LUO, AND Y. ZHANG, *A review of fault detection and diagnosis methods for power system*, IEEE Access, 7 (2019), pp. 162221–162239.

[26] Y. HAN, W. WANG, AND Y. ZHANG, *Stgnn: Spatial‚Äìtemporal graph neural network for multivariate time series anomaly detection*, Neurocomputing, 500 (2022), pp. 11–25.

[27] Q. HE, Y. ZHENG, C. ZHANG, AND H. WANG, *Mtad-tf: Multivariate time series anomaly detection using the combination of temporal pattern and feature pattern*, Complexity, 2020 (2020), p. Article ID 8846608.

[28] E. HODO, X. BELLEKENS, A. HAMILTON, P.-L. DUBOUILH, E. IORKYASE, C. TACHTATZIS, AND R. ATKINSON, *Threat analysis of iot networks using artificial neural network intrusion detection system*, in 2016 International Symposium on Networks, Computers and Communications (ISNCC), IEEE, 2016, pp. 1–6.

[29] K. HUNDMAN, V. CONSTANTINOU, C. LAPORTE, I. COLWELL, AND T. SODERSTROM, *Detecting space anomalies in spacecraft telemetry with lstms and nonparametric dynamic thresholding*, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, (2018).

[30] K. HUNDMAN, V. CONSTANTINOU, C. LAPORTE, I. COLWELL, AND T. SODERSTROM, *Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding*, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, (2018), pp. 387–395.

[31] Y. KIM, M.-G. KIM, AND U. KANG, *Learning spatial-temporal graph neural networks for multivariate time series anomaly detection*, in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 635–645.

[32] N. LAPTEV, S. AMIZADEH, AND I. FLINT, *Generic and scalable framework for automated time-series anomaly detection*, in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1939–1947.

[33] A. LAVIN AND S. AHMAD, *Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark*, in 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), IEEE, 2015, pp. 38–44.

[34] D. LI, D. CHEN, B. JIN, L. SHI, J. GOH, AND S.-K. NG, *Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks*, in Proc. Int. Conf. Artif. Neural Netw. (ICANN), 2019, pp. 703–716.

[35] Y. LI, R. YU, C. SHAHABI, AND Y. LIU, *Diffusion convolutional recurrent neural network: Data-driven traffic forecasting*, arXiv preprint arXiv:1707.01926, (2017).

[36] Z. LI, H. MA, D. CHEN, C. WU, AND H. XIE, *Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding*, in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2021, pp. 3220–3230.

[37] T. W. LIAO, *Clustering of time series data‚Äîa survey*, Pattern recognition, 38 (2005), pp. 1857–1874.

[38] F. T. LIU, K. M. TING, AND Z.-H. ZHOU, *Isolation forest*, in 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422.

[39] S. LLOYD, *Least squares quantization in pcm*, IEEE transactions on information theory, 28 (1982), pp. 129–137.

[40] W. LU, H. YAN, J. WANG, AND Y. LIU, *Transfer learning for process fault diagnosis: Knowledge transfer from simulation to physical processes*, Computers & Chemical Engineering, 134 (2020), p. 106675.

[41] L. V. D. MAATEN AND G. HINTON, *Visualizing data using t-sne*, Journal of machine learning research, 9 (2008), pp. 2579–2605.

[42] E. MANZOOR AND L. AKOGLU, *Fast memory-efficient anomaly detection in streaming heterogeneous graphs*, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1035–1044.

[43] A. P. MATHUR AND N. O. TIPPENHAUER, *Swat: A water treatment testbed for research and training on ics security*, in Proc. Int. Workshop Cyber-Phys. Syst. Security (CPSS), 2016, pp. 31–36.

[44] L. MCINNES, J. HEALY, AND J. MELVILLE, *Umap: Uniform manifold approximation and projection for dimension reduction*, in Proc. 2018 Int. Conf. on ArXiv Preprint, 2018, p. arXiv:1802.03426.

[45] F. MURTAGH AND P. CONTRERAS, *Algorithms for hierarchical clustering: an overview, ii*, Wiley interdisciplinary reviews. Data mining and knowledge discovery, 7 (2017), pp. e1219–n/a.

[46] G. PANG, C. SHEN, L. CAO, AND A. V. D. HENGEL, *Deep learning for anomaly detection: A review*, ACM Computing Surv., 54 (2021), pp. 1–38.

[47] E. SCHUBERT, J. SANDER, M. ESTER, H.-P. KRIEGEL, AND X. XU, *Dbscan revisited, revisited: Why and how you should (still) use dbscan*, in Proc. 2017 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2017, pp. 1087–1096.

[48] M.-L. SHYU, S.-C. CHEN, K. SARINNAPAKORN, AND L. CHANG, *A novel anomaly detection scheme based on principal component classifier*, in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, 2003, pp. 172–179.

[49] Y. SU, Y. ZHAO, C. NIU, R. LIU, W. SUN, AND J. PEI, *Robust anomaly detection for multivariate time series through stochastic recurrent neural network*, in

Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.

[50] Z. TIAN, L. SHI, S. LIU, Y. YU, H. WU, Q. HU, AND H. WANG, *Anomaly detection using spatial and temporal information in multivariate time series*, Sci. Rep., 13 (2023), p. 4400.

[51] F. WANG, Y. JIANG, R. ZHANG, A. WEI, J. XIE, AND X. PANG, *A survey of deep anomaly detection in multivariate time series: Taxonomy, applications, and directions*, Sensors, 25 (2025), p. 190.

[52] W. WU, P. CUI, AND J. PEI, *A comprehensive survey on graph neural networks*, IEEE Trans. Neural Netw. Learn. Syst., 32 (2021), pp. 4–24.

[53] Z. WU, S. PAN, F. CHEN, G. LONG, C. ZHANG, AND S. Y. PHILIP, *Graph wavenet for deep spatial-temporal graph modeling*, in IJCAI, 2019, pp. 1907–1913.

[54] Z. WU, S. PAN, F. CHEN, G. LONG, C. ZHANG, AND P. S. YU, *A comprehensive survey on graph neural networks*, IEEE Transactions on Neural Networks and Learning Systems, 32 (2021), pp. 4–24.

[55] F. XIA, K. SUN, S. YU, A. AZIZ, L. WAN, S. PAN, AND H. LIU, *Graph learning: A survey*, IEEE Trans. Artif. Intell., 2 (2021), pp. 109–127.

[56] S. YAN, Y. XIONG, AND D. LIN, *Spatial temporal graph convolutional networks for skeleton-based action recognition*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[57] Z. YANG, J. ZHANG, T. ZHANG, AND Y. SONG, *Spatial-temporal graph convolutional networks for power system fault detection*, IEEE Transactions on Smart Grid, 13 (2022), pp. 2759–2771.

[58] Z. YING, D. BOURGEOIS, J. YOU, M. ZITNIK, AND J. LESKOVEC, *Gnnexplainer: Generating explanations for graph neural networks*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 9240–9251.

[59] H. ZENATI, C. S. FOO, B. LECOUAT, G. MANEK, AND V. CHANDRASEKHAR, *Efficient gan-based anomaly detection*, in arXiv preprint arXiv:1802.06222, 2018.

[60] J. ZHANG, X. WANG, Y. YANG, H. MIAO, AND S. YANG, *A novel anomaly detection method for multivariate time series based on spatial-temporal graph learning*,

Journal of King Saud University - Computer and Information Sciences, 37 (2025), p. 9.

[61] J. ZHANG, Y. ZHENG, AND D. QI, *Deep spatio-temporal residual networks for citywide crowd flows prediction*, arXiv.org, (2017).

[62] H. ZHAO, Y. WANG, J. DUAN, C. HUANG, D. CAO, Y. TONG, B. XU, J. BAI, J. TONG, AND Q. ZHANG, *Multivariate time-series anomaly detection via graph attention network*, in Proc. 2020 IEEE Int. Conf. Data Mining (ICDM), 2020, pp. 841–850.

[63] Y. ZHENG, H. Y. KOH, M. JIN, L. CHI, K. T. PHAN, S. PAN, Y.-P. P. CHEN, AND W. XIANG, *Correlation-aware spatial-temporal graph learning for multivariate time-series anomaly detection*, IEEE Trans. Neural Netw. Learn. Syst., 35 (2024), pp. 11802–11816.

[64] J. ZHOU, G. CUI, Z. ZHANG, C. YANG, Z. LIU, M. SUN, AND W. LI, *Graph neural networks: A review of methods and applications*, AI Open, 1 (2020), pp. 57–81.