

Graph Neural Network based Spatio-Temporal Modelling for Anomaly Detection in Complex Network Systems

A THESIS

**Submitted to
Delhi Technological University**

**In Partial Fulfilment of the Requirement for the Award of
MASTER OF TECHNOLOGY DEGREE IN
Computer Science & Engineering**

SUBMITTED BY

**Dwiparna Mandal
(Roll No. 24/CSE/17)**

UNDER THE SUPERVISION OF:

Dr. Sanjay Kumar



**SCHOOL OF COMPUTER ENGINEERING
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi - 110042
June 2026**

DEPT. OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-11004

CANDIDATE'S DECLARATION

I, Dwiparna Mandal (24/CSE/17) students of M.Tech in Computer Science and Engineering, hereby declare that the Thesis titled “Graph Neural Network based Spatio-Temporal Modelling for Anomaly Detection in Complex Network Systems” which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, New Delhi in partial fulfillment of the requirement for awarding of the Master of Technology degree, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Fellowship or other similar title or recognition.

Place: New Delhi

Name: Dwiparna Mandal

Date:

Roll no.: 24/CSE/17

DEPT. OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that the project entitled

**Graph Neural Network based Spatio-Temporal Modelling for Anomaly
Detection in Complex Network Systems**

Submitted by

Dwiparna Mandal (24/CSE/17)

I hereby certify that the Thesis titled “Graph Neural Network based Spatio-Temporal Modelling for Anomaly Detection in Complex Network Systems” which is submitted by Dwiparna Mandal (24/CSE/17) for partial fulfillment of the requirements for awarding of the degree of Master of Technology (M.Tech) is a record of this work carried out by the students under my guidance & supervision. To the best of my knowledge, this work has not been submitted in any part or fulfillment for any Degree or Diploma to this University or elsewhere.

Date:

Dr. Sanjay Kumar
Assistant Professor
Dept. of Computer Science And Engineering
Delhi Technological University

Acknowledgements

The successful completion of any task would be incomplete without acknowledging the contributions of those whose support and guidance made it possible. This project would have remained an idea without their invaluable assistance and encouragement.

Firstly, I would like to extend my heartfelt gratitude to Prof. Anil Singh Parihar, Head of the Department, Computer Science and Engineering, Delhi Technological University, as well as the esteemed faculty members of our department. Their continuous guidance, motivation, and support have been instrumental throughout this project.

I am particularly thankful to my guide, Dr. Sanjay Kumar, for his unwavering encouragement and invaluable insights. His mentorship and supervision has played a vital role in shaping the idea and execution of this M.Tech Thesis. His support and availability for guidance at every step have been crucial to overcoming challenges along the way.

Additionally, I would like to express my sincere gratitude to everyone who directly or indirectly contributed to the completion of this project. Their assistance and encouragement have been a source of strength throughout this journey. Lastly, I extend my deepest thanks to my parents and well-wishers for their constant support, belief in me, and encouragement, which have been my foundation in pursuing this endeavor.

Dwiparna Mandal
Roll No.: 24/CSE/17

ABSTRACT

The high dynamics and relational nature of data generated by modern interconnected systems, such as financial transaction networks, IoT settings, communication networks, and cyber-physical systems, pose a significant challenge for anomaly detection. Such environments often have evolving graph structures, temporal dependencies and complex interaction patterns that cannot be captured by traditional machine learning approaches.

In this thesis, we propose GNN-based spatio-temporal frameworks for anomaly detection in dynamic network systems via three major contributions: DySTAL-AD, CaT-GNN and STARGATE. DySTAL-AD (Dynamic Self-Attention and LSTM-based Anomaly Detection) is an anomaly detection approach for IoT communication networks based on dynamic graph construction, structural and temporal self-attention, and LSTM refinement. We evaluate the framework on the CICIDS-2017, CICIDS-2018 and CICIOT-2022 datasets, and achieve better Precision, Recall, F1-score and Accuracy compared to GCN, GAT, GRU and DySAT baselines.

The second contribution, CaT-GNN [29] (Causal-Temporal Graph Neural Network), fuses causal reasoning with temporal graph learning to provide robust and interpretable fraud detection in dynamic financial transaction networks. To tackle the issue of temporal drift and spurious correlations, we propose a framework with Temporal Graph Encoder and Causal Inspector and Causal Intervener modules. Experiments on the Elliptic Bitcoin and PaySim datasets show improved AUC, F1-score, and interpretability compared to existing temporal and graph-based models.

The third work, STARGATE (Spatio-Temporal Anomaly Recognition with Graph Attention Networks and Transformer Encoder), is proposed for anomaly detection in satellite telemetry systems. We use Graph Attention Networks (GAT) to model the spatial dependencies and a Transformer encoder to learn long-range temporal dependencies in the telemetry sequences. Experimental results on NASA SMAP and MSL datasets show strong anomaly detection capability with high Precision, Recall and F1-score, compared to existing baseline and hybrid models.

Overall, the proposed frameworks demonstrate that the combination of graph representation learning, temporal modeling, self-attention mechanisms, recurrent refinement, transformer-based sequence learning, and causal reasoning can greatly improve the performance, robustness, and interpretability of anomaly detection in evolving complex network systems. This thesis provides scalable and efficient solutions for applications of cybersecurity, financial security, and satellite telemetry anomaly detection.

Keywords: Graph Neural Networks, Dynamic Graphs, Temporal Learning, IoT Security, Fraud Detection, Satellite Telemetry, Graph Attention Networks, Transformer Encoder, Dynamic Self-Attention, Causal Learning, Anomaly Detection.

Contents

CANDIDATE'S DECLARATION	I
CERTIFICATE	II
ACKNOWLEDGEMENT	III
ABSTRACT	IV
1 Introduction	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Research Objectives	2
1.4 Research Contributions	3
2 Literature Review	4
2.1 Traditional Anomaly Detection Approaches	4
2.2 Deep Learning Approaches for Sequential Modeling	4
2.3 Graph Neural Networks	5
2.4 Dynamic Graph Learning and Temporal Graph Neural Networks ..	6
2.5 Causal Representation Learning and Explainable Graph Learning ..	7
2.6 Research Gap and Motivation	7
3 Datasets and Preprocessing	9
3.1 Datasets	9
3.1.1 CICIDS-2017 Dataset	9
3.1.2 CICIDS-2018 Dataset	10
3.1.3 CICIoT-2022 Dataset	10
3.1.4 Elliptic Bitcoin Transaction Dataset	11
3.1.5 PaySim Mobile Money Transaction Dataset	11
3.1.6 NASA SMAP Dataset	11
3.1.7 NASA MSL Dataset	12
3.2 Data Preprocessing	13
3.2.1 Data Cleaning	13
3.2.2 Missing Value Handling	14
3.2.3 Feature Normalization	14
3.2.4 Categorical Feature Encoding	14
3.2.5 Dynamic Graph Construction	14
3.2.6 Temporal Sequence Construction	15
3.2.7 Class Imbalance Handling	15
3.2.8 Train-Test Splitting	15

3.2.9	Feature Extraction and Representation Learning	15
4	Proposed Methodology	17
4.1	DySTAL-AD: Dynamic Self-Attention and LSTM Model	17
4.1.1	Overview	17
4.1.2	Problem Formulation	18
4.1.3	Dynamic Graph Construction	19
4.1.4	Structural Self-Attention Module	20
4.1.5	Temporal Self-Attention Module	21
4.1.6	LSTM-Based Temporal Refinement	22
4.1.7	Binary Anomaly Classification	22
4.1.8	Loss Function	23
4.2	CaT-GNN [1]: Causal-Temporal Graph Neural Network	23
4.2.1	Overview	23
4.2.2	Dynamic Transaction Graph Representation	24
4.2.3	Temporal Graph Encoder	25
4.2.4	Causal Inspector Module	25
4.2.5	Causal Intervener Module	25
4.2.6	Fraud Classification	26
4.2.7	Overall Training Objective	26
4.3	STARGATE: Spatio-Temporal Anomaly Recognition with Graph	
	Attention Networks and Transformer Encoder	27
4.3.1	Overview	27
4.3.2	Problem Formulation	28
4.3.3	Telemetry Sequence Construction	28
4.3.4	Graph Attention Network Module	29
4.3.5	Transformer-Based Temporal Learning	30
4.3.6	Anomaly Classification	31
4.3.7	Loss Function	31
5	Experimental Results and Discussion	32
5.1	Experimental Setup	32
5.2	Evaluation Metrics	33
5.2.1	Accuracy	33
5.2.2	Precision	33
5.2.3	Recall	33
5.2.4	F1-score	33
5.2.5	AUC-ROC	34
5.3	Results of DySTAL-AD Model	34

5.3.1	Results on CICIDS-2017 Dataset	34
5.3.2	Results on CICIDS-2018 Dataset	34
5.3.3	Results on CICIoT-2022 Dataset	35
5.3.4	Accuracy Comparison Across Datasets	36
5.4	Results of CaT-GNN model	37
5.4.1	AUC Performance Comparison	37
5.4.2	Precision Performance Comparison	38
5.4.3	F1-score Performance Comparison	38
5.4.4	Discussion	39
5.5	Experimental Results of STARGATE	39
5.5.1	Comparative Performance Analysis	40
5.5.2	Effect of Window Length	42
5.5.3	Training Loss Analysis for Hidden Nodes	42
5.5.4	Training Loss Analysis for Transformer Layers	43
5.5.5	Validation Loss Analysis for Hidden Nodes	44
5.5.6	Validation Loss Analysis for Transformer Layers	44
5.5.7	Discussion	45
5.6	Comparative Discussion	45
6	Conclusion and Future Work	47

List of Figures

4.1	Architecture of the Proposed DySTAL-AD model	18
4.2	Architecture of the Proposed CaT-GNN model	24
4.3	Architecture of the Proposed STARGATE model	27
5.1	Accuracy Comparison of Different Models Across CICIDS-2017, CICIDS-2018, and CICIoT-2022 Datasets	36
5.2	Cross Model AUC Comparison for PaySim and Elliptic Datasets	37
5.3	Cross Model Precision Comparison for Elliptic and PaySim Datasets	38
5.4	Cross Model F1-Score Comparison for Elliptic and PaySim Datasets	39
5.5	F1-score Comparison Across Different Models	40
5.6	Precision Comparison Across Different Models	41
5.7	Recall Comparison Across Different Models	41
5.8	F1-score and Training Time versus Window Length	42
5.9	Training Loss versus Number of Hidden Nodes	43
5.10	Training Loss versus Number of Transformer Layers	43
5.11	Validation Loss versus Number of Hidden Nodes	44
5.12	Validation Loss versus Number of Transformer Layers	45

List of Tables

2.1	Comparison of Existing Models and Research Gaps	8
3.1	Summary of Datasets Used in the Thesis	13
5.1	Performance results on CICIDS-2017	34
5.2	Performance Comparison on CICIDS-2018	35
5.3	Performance Comparison on CICIoT-2022	35

Chapter 1

INTRODUCTION

1.1 Background

The fast evolution of connected systems has transformed the contemporary digital infrastructure. Internet of Things (IoT) devices, financial transaction systems, communication networks, satellite telemetry systems and cyber-physical infrastructures continuously generate massive volumes of highly interconnected and time-dependent data [2-5]. Such systems can be naturally represented as graphs where nodes are entities and edges are interactions between them. These environments are prone to anomalous activities such as cyber intrusions, distributed attacks, fraudulent financial transactions, malicious communications, and telemetry failures that often manifest themselves in the form of abnormal structural and temporal interaction patterns. The detection of such anomalies is key to ensure the security, reliability, operational stability and resilience of the system.

Traditional anomaly detection methods are largely based on hand-crafted statistical features and shallow machine learning algorithms [6,7]. These techniques work reasonably well in small-scale or static environments but fail to capture complex relational dependencies and temporal evolution in modern dynamic systems. Recent advances in deep learning have introduced recurrent architectures such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU) that effectively model temporal dependencies in sequential data [8,9]. These methods, however, treat the observations independently and do not explicitly learn the relational graph structures.

Graph Neural Networks (GNNs) have emerged as powerful architectures for learning on graph-structured data. Models such as Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and GraphSAGE effectively capture structural relationships using neighborhood aggregation mechanisms [10-12]. Nevertheless, most existing GNN architectures assume static graph topology and

cannot adequately model evolving interactions over time. Dynamic graph learning addresses this limitation by representing systems as sequences of evolving graph snapshots [13–17]. Furthermore, Transformer-based temporal learning approaches have shown strong capability in modeling long-range temporal dependencies in sequential and graph-based systems [18–20]. In parallel, causal reasoning has emerged as an important research direction for improving model robustness and interpretability by disentangling causal relationships from spurious correlations [21–24].

1.2 Problem Statement

Modern complex network systems such as IoT environments, communication infrastructures, satellite telemetry systems, and financial transaction networks generate massive volumes of interconnected and time-dependent data, where anomalous activities are often hidden within evolving structural and temporal interaction patterns [2–5, 25]. Traditional anomaly detection approaches based on statistical methods or conventional machine learning techniques fail to effectively capture dynamic topology evolution, relational dependencies among entities, and long-term behavioral changes. Although Graph Neural Networks (GNNs) can model structural relationships and recurrent architectures such as LSTM and GRU can capture temporal dependencies, existing approaches generally focus on either structural or temporal learning independently and lack a unified model for jointly modeling both aspects. Furthermore, most existing models suffer from challenges such as class imbalance, poor interpretability, sensitivity to spurious correlations, and limited robustness under evolving network conditions. Existing methods for telemetry anomaly detection also face difficulty in modeling long-range temporal dependencies and complex spatial interactions among telemetry channels [4, 5, 20]. Therefore, there is a need to develop unified spatio-temporal graph learning models capable of integrating structural representation learning, temporal evolution modeling, Transformer-based sequence learning, and causal reasoning to accurately, robustly, and efficiently detect anomalies in dynamic complex network systems.

1.3 Research Objectives

The main objective of this thesis is to propose robust and scalable spatio-temporal models based on Graph Neural Networks for anomaly detection in dynamic complex network systems. The research focuses on dynamic graph learning for modeling the evolution of structural and temporal dependencies in IoT communication networks, financial transaction systems, and satellite telemetry environments. The

this thesis aims to combine structural self-attention, temporal self-attention, recurrent memory refinement, Transformer-based temporal learning, and causal reasoning mechanisms to improve anomaly detection accuracy, robustness, and interpretability. Furthermore, the work addresses important challenges including topology evolution, long-term behavioral modeling, temporal drift, class imbalance, spurious correlations, and dynamic dependency learning. The proposed models are evaluated on benchmark intrusion detection, fraud detection, and telemetry anomaly detection datasets to demonstrate their effectiveness compared to existing state-of-the-art graph-based and temporal learning approaches [14, 15, 19, 20].

1.4 Research Contributions

The major contributions of this thesis are centered around the development of advanced spatio-temporal and causal graph learning models for anomaly detection in dynamic network systems. First, a Dynamic Self-Attention and LSTM-based model named DySTAL-AD is proposed for anomaly detection in evolving IoT communication networks, where dynamic graph construction, structural self-attention, temporal self-attention, and recurrent memory refinement are jointly integrated to capture topology evolution and persistent anomalous behavior. Second, we propose a Causal-Temporal Graph Neural Network (CaT-GNN) [1] which combines the temporal graph representation learning and causal reasoning for robust and interpretable fraud detection in dynamic [1] financial transaction networks by intervention-based debiasing and causal attention mechanisms [1]. Third, a hybrid model named STARGATE is proposed for anomaly detection in satellite telemetry systems by integrating Graph Attention Networks with Transformer encoders to jointly model spatial telemetry dependencies and long-range temporal dynamics [11, 18, 20]. The proposed models demonstrate superior or highly competitive performance on benchmark datasets including CICIDS-2017, CICIDS-2018, CICIoT-2022, Elliptic, PaySim, NASA SMAP, and NASA MSL datasets [2, 5, 25]. The thesis further establishes that integrating graph representation learning, temporal modeling, self-attention mechanisms, recurrent refinement, Transformer-based sequence learning, and causal reasoning significantly improves anomaly detection robustness, generalization capability, scalability, and interpretability in evolving complex network environments.

Chapter 2

LITERATURE REVIEW

2.1 Traditional Anomaly Detection Approaches

Early anomaly detection approaches were based on statistical analysis and traditional machine learning algorithms to detect abnormal patterns in network systems and transactional environments. Methods like Support Vector Machines (SVM), Random Forests, Decision Trees, Logistic Regression, and k-Nearest Neighbors (kNN) were widely used for intrusion detection and fraud analysis because of their simplicity and relatively low computational complexity [6,7]. Such methods were usually based on manually designed statistical features from traffic flows or transaction logs, like the packet counts, flow duration, byte statistics, transaction amount, and behavioral frequencies. These methods performed reasonably well in small-scale, relatively static environments but had significant limitations when applied to modern complex network systems. Most traditional techniques analyzed samples independently and failed to capture relational dependencies between communicating entities or transactional participants. Moreover, they could not effectively model evolving temporal behavior, topology changes, and long-term sequential dependencies. Consequently, these methods struggled to generalize in highly dynamic environments such as IoT networks, communication systems, financial transaction graphs, and telemetry systems, where anomalies often arise from coordinated or evolving interactions rather than isolated events [2-5].

2.2 Deep Learning Approaches for Sequential Modeling

With the advancement of deep learning, powerful architectures were introduced to automatically learn hierarchical feature representations from large-scale data. Re-

current neural network architectures such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU) became widely adopted in anomaly detection research for modeling sequential and temporal patterns in network traffic, telemetry streams, and financial transactions [8,9]. LSTM networks addressed the vanishing gradient problem using memory cells and gating mechanisms, enabling effective learning of long-term temporal dependencies. GRU provided a computationally efficient alternative with similar temporal learning capability. Recurrent models significantly outperformed traditional machine learning methods in modeling communication flow dynamics, attack progression, transaction behavior, and telemetry sequence anomalies [4,5,26]. These approaches were successfully applied in cybersecurity applications to detect emerging attack patterns and in financial systems for transaction fraud analysis. However, recurrent architectures primarily operated on vectorized sequential data and ignored the relational graph structure among interacting entities. As a result, graph-based dependencies such as communication topology, neighborhood interactions, and collaborative anomalous behavior could not be explicitly modeled. This limitation motivated the integration of graph representation learning with temporal modeling for more comprehensive anomaly detection models.

2.3 Graph Neural Networks

Graph Neural Networks (GNNs) have become an effective method of learning over graph-structured data by explicitly learning the relationship among connected entities. In network security and fraud detection systems, nodes may be devices, users, accounts, telemetry channels or transactions, and edges may be communications, transfers or behavioral relationships. The Graph Convolutional Networks (GCN) proposed by Kipf and Welling became one of the basic architectures for graph learning [10]. GCN generalized the convolution operations from Euclidean domains to non-Euclidean graph domains through the neighborhood aggregation mechanisms, which can effectively learn the spatial dependencies among connected nodes. Graph Attention Networks (GAT) [11] further improved the learning of graph representations by adding attention mechanisms to give adaptive importance weights to neighboring nodes. This, in turn, allowed the model to give more weight to influential neighbors and improved the quality of the representation. GraphSAGE further pushed the scalability of GNNs by introducing inductive neighborhood sampling and aggregation, which enables efficient learning on large graphs [12]. These graph learning methods greatly enhanced the power of anomaly detection by learning the structural relations among entities. However, most existing GNN architectures assumed static graph topology and did not consider

temporal evolution in dynamic systems. In real-world environments, e.g., IoT systems, transaction networks and satellite telemetry infrastructures, communication patterns are ever-changing and the relational structures are dynamically evolving. Thus, the assumption of static graphs constrains the effectiveness and robustness of traditional GNN-based anomaly detection models.

2.4 Dynamic Graph Learning and Temporal Graph Neural Networks

To overcome the limitations of static graph learning, researchers introduced dynamic graph learning methods capable of modeling evolving graph structures over time. Dynamic graph representation learning treats systems as sequences of temporal graph snapshots or continuous-time interaction streams, enabling the modeling of topology evolution, interaction drift, and temporal dependencies. EvolveGCN extended traditional GCN architectures by dynamically evolving graph convolution parameters using recurrent neural networks [13]. Temporal Graph Networks (TGN) introduced memory-based architectures for learning continuous-time graph interactions through temporal message passing and event-driven updates [14]. Self-attention-based dynamic graph models such as Dynamic Self-Attention Networks (DySAT) integrated structural self-attention and temporal self-attention to jointly capture graph structure and temporal evolution without relying entirely on recurrent processing [15]. Transformer-based temporal graph architectures further improved dynamic representation learning by modeling long-range temporal dependencies using self-attention mechanisms [18-20]. Recent studies also explored graph-transformer hybrid models for telemetry anomaly detection and spatio-temporal learning in complex systems [27,28]. These methods achieved significant performance gains in evolving network environments by effectively modeling dynamic topology changes and temporal interaction patterns. However, several challenges still remain, including limited capability for long-term anomaly persistence modeling, computational complexity in large-scale dynamic graphs, lack of interpretability, and vulnerability to noise and spurious correlations. Furthermore, many dynamic graph models remain correlation-driven and lack mechanisms for causal reasoning and robust generalization under temporal distribution shifts.

2.5 Causal Representation Learning and Explainable Graph Learning

Causal representation learning has recently emerged as an important research direction for improving robustness, interpretability, and generalization of deep learning systems. Conventional machine learning and graph learning models primarily learn statistical correlations instead of true causal relationships, making them vulnerable to spurious patterns, distribution shifts, and adversarial conditions. In anomaly detection applications, this limitation often leads to unstable predictions and poor generalization when system behavior evolves over time. To address these challenges, causal learning models attempt to distinguish causally relevant features from non-causal correlations using intervention-based reasoning, invariant learning, and structural causal modeling [21, 22, 29]. Techniques such as Invariant Risk Minimization (IRM) improve robustness by enforcing consistency across multiple environments. In graph learning, CausalGNN and related approaches introduced causal regularization mechanisms to reduce spurious neighborhood influence and improve stable prediction capability [23]. More recent temporal causal graph models integrated dynamic graph learning with causal reasoning to improve robustness under evolving graph conditions [24]. Explainable graph learning methods additionally aim to identify important nodes, edges, and subgraphs responsible for model predictions, thereby improving interpretability in security-critical applications such as fraud detection, intrusion detection, and telemetry anomaly analysis. Nevertheless, most existing causal graph learning approaches remain limited to static graphs or simplified temporal settings. A major research gap therefore exists in developing unified models that jointly incorporate dynamic graph learning, temporal modeling, Transformer-based sequence learning, causal reasoning, and interpretability for anomaly detection in evolving complex network systems.

2.6 Research Gap and Motivation

Based on the comprehensive literature survey, several important research gaps have been identified in existing anomaly detection models for dynamic network systems. Traditional machine learning approaches cannot effectively model structural and temporal dependencies, while recurrent neural architectures lack the capability to represent relational graph structures. Existing graph neural networks effectively capture spatial dependencies but generally assume static graph topology, limiting their applicability in evolving environments. Dynamic graph learning methods improve temporal representation capability but often fail to capture

long-term anomaly persistence and remain vulnerable to spurious correlations and temporal distribution shifts. Furthermore, interpretability and causal reasoning remain under-explored in dynamic graph-based anomaly detection systems. Existing telemetry anomaly detection methods also face limitations in jointly modeling spatial dependencies and long-range temporal evolution [4, 5, 20]. Most existing approaches focus independently on structural learning, temporal modeling, or causal reasoning rather than integrating all aspects into a unified model. These challenges motivate this thesis to develop advanced spatio-temporal and causal graph learning models that jointly integrate dynamic graph construction, structural self-attention, temporal modeling, recurrent refinement, Transformer-based sequence learning, and causal reasoning to improve robustness, interpretability, scalability, and anomaly detection performance in evolving complex network environments.

Table 2.1 compares existing anomaly detection approaches with respect to structural learning, temporal modeling, dynamic graph capability, causal reasoning, and interpretability.

Table 2.1: Comparison of Existing Models and Research Gaps

Model	Structural	Temporal	Dynamic	Causal	Interpretability	Major Limitation
Traditional ML	No	No	No	No	Low	Cannot model relational and temporal dependencies.
LSTM / GRU	No	Yes	No	No	Low	Ignores graph structure and entity relationships.
GCN	Yes	No	No	No	Moderate	Assumes static graph topology.
GAT	Yes	No	No	No	Moderate	Cannot capture temporal evolution.
GraphSAGE	Yes	Partial	Partial	No	Moderate	Weak temporal dependency learning.
EvolveGCN	Yes	Yes	Yes	No	Moderate	Limited interpretability and memory modeling.
TGN	Yes	Yes	Yes	No	Moderate	High computational complexity.
DySAT	Yes	Yes	Yes	No	Moderate	Lacks causal reasoning capability.
Transformer-based Models	Partial	Yes	Partial	No	Moderate	Weak relational graph representation capability.
CausalGNN	Yes	Partial	No	Yes	High	Limited support for dynamic temporal graphs.
Telemetry Anomaly Models	No	Yes	No	No	Moderate	Cannot jointly model spatial and temporal dependencies.
DySTAL-AD (Proposed)	Yes	Yes	Yes	Partial	High	Improves temporal refinement and anomaly persistence detection.
CaT-GNN (Proposed)	Yes	Yes	Yes	Yes	High	Integrates causal reasoning with temporal graph learning.
STARGATE (Proposed)	Yes	Yes	Yes	Partial	High	Integrates GAT and Transformer for telemetry anomaly detection.

Existing methods fail to jointly capture dynamic graph evolution, temporal dependencies, causal relationships, and interpretability. These limitations motivate the proposed DySTAL-AD, CaT-GNN [1], and STARGATE models for robust spatio-temporal anomaly detection.

Chapter 3

DATASETS AND PREPROCESSING

3.1 Datasets

The proposed spatio-temporal anomaly detection models are evaluated using benchmark datasets from cybersecurity, financial fraud detection, and satellite telemetry domains. These datasets were selected because they contain realistic network traffic, dynamic interactions, temporal dependencies, telemetry sequences, and diverse attack or anomaly scenarios. The datasets used in this thesis include CICIDS-2017, CICIDS-2018, CICIoT-2022, Elliptic Bitcoin Transaction Dataset, PaySim Mobile Money Transaction Dataset, NASA SMAP, and NASA MSL datasets [2, 5, 25]. Together, these datasets provide a comprehensive evaluation environment for analyzing the robustness, scalability, interpretability, and generalization capability of the proposed DySTAL-AD, CaT-GNN [1], and STARGATE models under different real-world conditions.

3.1.1 CICIDS-2017 Dataset

The CICIDS-2017 dataset is one of the most commonly used benchmark datasets research [2]. It was developed by the Canadian Institute for Cybersecurity (CIC) and contains realistic enterprise network traffic and multiple modern attack scenarios. The dataset contains benign and malicious traffic under realistic user behaviour profiles. Attack types: DDoS, DoS, PortScan, Brute Force, Botnet activity, infiltration attacks, web attacks.

Each traffic flow contains statistical and protocol-based features extracted using CICFlowMeter, including packet count, byte rate, flow duration, source and destination ports, protocol information, and inter-arrival time statistics. The

evolving communication behavior and temporal attack patterns make the dataset highly suitable for dynamic graph-based anomaly detection. In this thesis, the dataset is used to evaluate the DySTAL-AD model for binary anomaly detection.

3.1.2 CICIDS-2018 Dataset

The CICIDS-2018 dataset is an extended and more comprehensive version of CICIDS-2017 [3]. It contains large-scale enterprise-level network traffic with diverse cyberattack scenarios including brute-force attacks, botnet communication, DDoS attacks, infiltration attacks, and malicious communication behavior.

Compared to CICIDS-2017, the dataset contains larger traffic volume, higher diversity, and more complex temporal communication patterns. It also includes highly imbalanced attack distributions, making anomaly detection significantly more challenging. The dynamic topology evolution and heterogeneous communication patterns make this dataset highly suitable for evaluating spatio-temporal graph learning models. In this work, CICIDS-2018 is used to evaluate the robustness and scalability of DySTAL-AD under large-scale network environments.

3.1.3 CICIOT-2022 Dataset

The CICIOT-2022 dataset is specifically designed for IoT security and anomaly detection research. It contains communication traffic generated from heterogeneous IoT devices operating in dynamic environments. The dataset includes both normal communication and multiple attack scenarios such as DDoS attacks, spoofing attacks, reconnaissance activities, and malicious device interactions.

IoT environments involve constrained devices, dynamic topology evolution, heterogeneous communication protocols, and highly variable traffic behavior. The dataset captures these characteristics through continuously evolving communication patterns among IoT devices. Each traffic instance contains flow-level features such as packet statistics, communication frequency, protocol information, and behavioral attributes.

The evolving interaction patterns naturally form dynamic graph structures over time, making the dataset highly suitable for evaluating graph-based temporal anomaly detection models. In this thesis, CICIOT-2022 is used to evaluate the capability of DySTAL-AD in modeling topology drift and long-term anomalous communication behavior.

3.1.4 Elliptic Bitcoin Transaction Dataset

The Elliptic Bitcoin Transaction Dataset is a large-scale cryptocurrency transaction graph dataset designed for anti-money laundering and fraud detection research [25]. The dataset represents Bitcoin transactions as graph structures where nodes correspond to transactions and edges represent Bitcoin flow between transactions. The dataset contains 49 temporal graph snapshots collected over time, making it highly suitable for temporal graph learning and dynamic fraud detection.

Each transaction node contains 203 handcrafted features describing transactional behavior, temporal activity, and structural properties. Transactions are labeled as licit, illicit, or unknown. Illicit transactions are associated with fraudulent activities such as money laundering and suspicious financial operations.

The dynamic and interconnected nature of cryptocurrency transactions makes the dataset ideal for evaluating temporal graph neural networks and causal reasoning models. In this thesis, the dataset is used to evaluate the proposed CaT-GNN [1] model for accurate and understandable fraud detection.

3.1.5 PaySim Mobile Money Transaction Dataset

PaySim is a synthetic financial transaction dataset generated using real-world mobile money transaction logs collected from an African mobile payment platform. The dataset simulates realistic mobile money behavior while preserving user privacy. It contains millions of transaction records representing activities such as cash transfer, payment, debit, cash-out, and account transfer operations.

Each transaction contains attributes including transaction amount, transaction type, account balance information, timestamp details, and fraud labels. Fraudulent transactions simulate real-world financial fraud scenarios including suspicious transfers and unauthorized account activities.

The dataset is highly imbalanced because fraudulent transactions represent only a very small portion of the total transaction volume. Additionally, transaction behavior evolves over time, creating temporal dependencies and distribution shifts. These characteristics make PaySim suitable for evaluating the robustness and scalability of the proposed CaT-GNN [1] model under dynamic financial environments.

3.1.6 NASA SMAP Dataset

The NASA Soil Moisture Active Passive (SMAP) dataset is a satellite telemetry dataset widely used for anomaly detection research in spacecraft systems [4].

The dataset contains multivariate telemetry sequences collected from spacecraft sensors and monitoring channels. Each telemetry channel records continuous temporal measurements related to satellite subsystem operations and environmental conditions.

The dataset contains both normal operational sequences and anomalous telemetry behavior caused by system faults, communication disturbances, and abnormal spacecraft conditions. The highly correlated telemetry channels and long temporal dependencies make anomaly detection particularly challenging.

In this thesis, the SMAP dataset is used to evaluate the proposed STARGATE model for spatio-temporal anomaly detection in satellite telemetry systems.

3.1.7 NASA MSL Dataset

The NASA Mars Science Laboratory (MSL) dataset is another benchmark telemetry anomaly detection dataset containing spacecraft telemetry sequences collected from the Curiosity rover mission [5]. Similar to SMAP, the dataset contains multivariate temporal sensor measurements collected from multiple spacecraft subsystems.

The dataset includes anomalous telemetry behavior caused by operational faults, communication irregularities, and abnormal subsystem activities. The telemetry channels exhibit strong temporal dependencies and inter-channel correlations, making the dataset suitable for evaluating advanced spatio-temporal learning models.

In this thesis, the effectiveness of the STARGATE model in modeling spatial telemetry relationships and long-range temporal dependencies for anomaly detection in satellite systems is evaluated on the MSL dataset.

Table 3.1: Summary of Datasets Used in the Thesis

Dataset	Domain	Type	Temporal Nature	Major Attacks / Anomalies	Purpose in Thesis
CICIDS-2017	Cybersecurity	Network Traffic	Dynamic	DoS, DDoS, PortScan, Brute Force, Botnet	Evaluation of DySTAL-AD for intrusion and anomaly detection.
CICIDS-2018	Cybersecurity	Enterprise Network Traffic	Dynamic	Web attacks, Botnet, DDoS, Infiltration	Testing robustness and scalability under large-scale traffic environments.
CICIoT-2022	IoT Security	IoT Communication Traffic	Highly Dynamic	DDoS, Spoofing, Reconnaissance	Evaluation of dynamic topology learning in heterogeneous IoT environments.
Elliptic Bitcoin Dataset	Financial Security	Transaction Graph	Temporal Graph Snapshots	Illicit Bitcoin Transactions	Evaluation of CaT-GNN for fraud and money laundering detection.
PaySim Dataset	Financial Security	Mobile Money Transactions	Dynamic	Fraudulent Money Transfers	Testing causal-temporal fraud detection capability.
NASA SMAP Dataset	Satellite Systems	Telemetry Time Series	Sequential Temporal	Sensor Failures, Telemetry Anomalies	Evaluation of STAR-GATE for satellite telemetry anomaly detection.
NASA MSL Dataset	Satellite Systems	Telemetry Time Series	Sequential Temporal	Operational Faults, Abnormal Telemetry Behavior	Testing long-range temporal and spatial anomaly modeling capability.

3.2 Data Preprocessing

Data preprocessing is an essential step to improve the quality, stability, and effectiveness of deep learning models. Since the datasets used in this thesis contain noisy records, missing values, categorical attributes, redundant information, temporal inconsistencies, and severe class imbalance, multiple preprocessing operations are performed prior to model training and evaluation.

3.2.1 Data Cleaning

The raw datasets initially contain duplicate records, corrupted entries, incomplete communication flows, and inconsistent feature values. Such noisy samples can negatively affect the learning process and degrade model performance. Therefore, duplicate traffic flows, invalid transaction records, and corrupted telemetry sequences are removed during preprocessing. Furthermore, samples containing undefined or infinite feature values are filtered to ensure data consistency and stable model optimization.

3.2.2 Missing Value Handling

Several datasets contain missing or undefined feature values due to incomplete network communication, sensor failures, or transaction logging errors. Missing values can adversely affect model convergence and produce unstable learning behavior. To address this issue, missing numerical values are handled using zero imputation or statistical replacement methods such as mean substitution. This ensures consistent feature dimensionality across all samples and improves numerical stability during training.

3.2.3 Feature Normalization

The datasets contain features with significantly different numerical ranges. Features such as packet counts, transaction amounts, telemetry sensor values, byte rates, and communication durations may vary substantially in magnitude. Large feature variations can negatively influence gradient-based optimization and bias the learning process toward dominant features. Therefore, Min-Max normalization is applied to scale numerical features into the range $[0, 1]$. The normalization process is defined as:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.1)$$

where x represents the original feature value and x' denotes the normalized feature value. Feature normalization improves convergence speed, training stability, and numerical robustness during optimization.

3.2.4 Categorical Feature Encoding

Some of the datasets contain categorical attributes such as protocol type, transaction category, service type, communication flags and telemetry channel identifiers. Since deep learning architectures require numerical input representations, categorical features are transformed using one-hot encoding. This process converts categorical variables into binary vectors while preserving category-specific information. One-hot encoding prevents the model from incorrectly interpreting categorical variables as ordinal numerical values.

3.2.5 Dynamic Graph Construction

To model evolving interactions, traffic flows, transaction records, and telemetry observations are segmented into temporal windows using a sliding window mechanism. For each temporal window, graph snapshots are constructed where nodes

represent entities such as devices, users, transactions, or telemetry channels, while edges represent interactions or relational dependencies among them. Node features are generated by aggregating statistical communication, transactional, or telemetry attributes within each temporal window.

The dynamic graph construction process enables the proposed models to capture topology evolution, interaction drift, temporal dependencies, and spatial correlations [13, 15, 17]. The resulting graph sequences serve as inputs to the proposed spatio-temporal and causal graph learning architectures.

3.2.6 Temporal Sequence Construction

For telemetry anomaly detection, sequential telemetry observations are segmented into fixed-length temporal sequences using sliding temporal windows. Each sequence captures long-range temporal dependencies among telemetry channels over time. This preprocessing strategy enables Transformer-based architectures to effectively learn temporal patterns, contextual dependencies, and anomalous telemetry behavior [4, 5, 18, 20].

3.2.7 Class Imbalance Handling

Since anomalous events are rare compared to normal behavior, datasets for anomaly detection are usually highly imbalanced. This imbalance may bias the model toward majority classes and decrease the sensitivity of anomaly detection. To mitigate this issue, weighted loss functions are utilized during training. Higher weights are assigned to anomalous samples to increase their contribution during optimization. Weighted binary cross-entropy and causal regularization losses are employed to improve detection performance for minority anomaly classes.

3.2.8 Train-Test Splitting

Datasets are divided into training, validation, and testing subsets using stratified splitting strategies. As a rule of thumb, 70% of the data is used for training, 15% for validation and 15% for testing. Stratified sampling ensures that the distribution of normal and anomalous samples remains consistent across all subsets. This prevents biased evaluation results and provides reliable performance estimation under highly imbalanced conditions.

3.2.9 Feature Extraction and Representation Learning

The normalized feature vectors are converted into graph-based and sequential representations suitable for deep learning. Graph adjacency matrices represent

structural relationships among entities, while temporal graph snapshots capture dynamic interaction evolution over time. For telemetry datasets, temporal sequences preserve long-range dependencies among sensor channels. These processed representations form the foundation of the proposed DySTAL-AD, CaT-GNN [1], and STARGATE models for learning spatio-temporal and causal anomaly patterns effectively [10, 11, 15, 18, 20, 23].

Chapter 4

PROPOSED METHODOLOGY

This chapter presents the proposed spatio-temporal graph learning models developed for anomaly detection in dynamic complex network systems. The methodologies are designed to jointly model structural dependencies, temporal evolution, dynamic topology changes, long-range sequential behavior, and causal relationships using Graph Neural Networks (GNNs), self-attention mechanisms, recurrent learning, Transformer-based temporal modeling, and causal reasoning techniques [8, 11, 15, 18, 23]. The chapter consists of three major models: DySTAL-AD for IoT anomaly detection, CaT-GNN [1] for robust and interpretable fraud detection, and STARGATE for satellite telemetry anomaly detection. Dynamic graph snapshots are constructed using temporal sliding windows where nodes represent entities and edges represent interactions among them. Structural and temporal dependencies are learned using graph aggregation and attention mechanisms, while recurrent and Transformer architectures capture long-term temporal behavior. The proposed models are evaluated on benchmark datasets including CICIDS-2017, CICIDS-2018, CICIOT-2022, Elliptic Bitcoin, PaySim, NASA SMAP, and NASA MSL datasets [2, 5, 25]. Experimental results demonstrate improved robustness, interpretability, scalability, and anomaly detection performance in evolving complex network environments.

4.1 DySTAL-AD: Dynamic Self-Attention and LSTM Model

4.1.1 Overview

DySTAL-AD is a Dynamic Self-Attention and LSTM-based model designed for binary anomaly detection in evolving IoT communication networks. The model combines dynamic graph construction, structural self-attention, temporal self-

attention, and recurrent memory refinement to effectively model evolving communication patterns and persistent anomalous behavior.

The overall architecture consists of the following stages:

1. Dynamic graph construction
2. Structural self-attention learning
3. Temporal self-attention learning
4. LSTM-based temporal refinement
5. Binary anomaly classification

The proposed model converts network traffic into a sequence of dynamic graph snapshots and learns topology-aware temporal representations for robust anomaly detection.

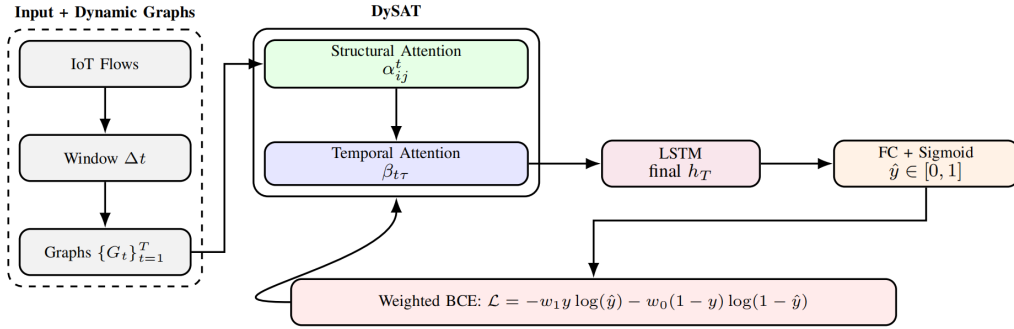


Figure 4.1: Architecture of the Proposed DySTAL-AD model

Figure 4.1 illustrates the overall architecture of the proposed DySTAL-AD model. The model first converts IoT traffic flows into dynamic graph snapshots using sliding temporal windows. Structural and temporal self-attention mechanisms are then applied using the DySAT model to capture evolving graph dependencies and temporal interactions. The generated embeddings are further refined using an LSTM layer to model long-term anomalous behavior, followed by a fully connected classification layer with sigmoid activation for anomaly prediction.

4.1.2 Problem Formulation

Let the raw IoT traffic flows be represented as:

$$X = \{x_1, x_2, x_3, \dots, x_N\} \quad (4.1)$$

where each traffic flow x_i contains protocol-level and statistical communication features such as packet count, byte rate, duration, source IP, destination IP, and communication flags.

The traffic is segmented into temporal windows to generate a sequence of dynamic graph snapshots:

$$G = \{G_1, G_2, G_3, \dots, G_T\} \quad (4.2)$$

where each graph snapshot at time t is defined as:

$$G_t = (V_t, E_t) \quad (4.3)$$

Here:

- V_t denotes the set of active devices or entities.
- E_t represents communication interactions between entities.

The objective of the model is to learn a mapping function:

$$f : G \rightarrow \{0, 1\} \quad (4.4)$$

where:

- 0 represents benign behavior.
- 1 represents anomalous behavior.

4.1.3 Dynamic Graph Construction

The first stage of DySTAL-AD involves transforming raw traffic flows into dynamic graph representations. Since communication patterns evolve continuously over time, static graph representations are insufficient for capturing topology drift and temporal interaction changes.

The traffic flows are segmented using sliding temporal windows of duration Δt . For each time window, a graph snapshot is constructed as:

$$G_t = (V_t, E_t) \quad (4.5)$$

where nodes correspond to communicating devices and edges represent interactions between devices during the time interval.

Each node is assigned a feature vector:

$$x_i^t = [f_1, f_2, f_3, \dots, f_d] \quad (4.6)$$

where the feature vector may include:

- Packet count
- Byte rate
- Flow duration
- Protocol type
- Connection frequency
- Communication statistics

The dynamic graph sequence:

$$G = \{G_1, G_2, \dots, G_T\} \quad (4.7)$$

captures evolving communication behavior and topology changes over time.

4.1.4 Structural Self-Attention Module

The structural self-attention module captures spatial dependencies among connected entities within each graph snapshot. Instead of assigning equal importance to neighboring nodes, the attention mechanism learns adaptive importance weights based on node interactions.

For each node i and its neighboring node j , the attention coefficient is computed as:

$$\alpha_{ij}^t = \frac{\exp(\text{LeakyReLU}(a^T[Wh_i^t || Wh_j^t]))}{\sum_{k \in N(i)} \exp(\text{LeakyReLU}(a^T[Wh_i^t || Wh_k^t]))} \quad (4.8)$$

where:

- W is the learnable weight matrix.
- a is the attention vector.
- $||$ denotes concatenation.
- $N(i)$ represents neighboring nodes.

The normalized attention score α_{ij}^t indicates the relative importance of neighbor j for node i .

The node embedding update is computed as:

$$h_i^{t,struct} = \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^t W h_j^t \right) \quad (4.9)$$

where $\sigma(\cdot)$ denotes the nonlinear activation function.

To improve representation stability, multi-head attention is employed:

$$h_i^{t,struct} = \left\|_{m=1}^M \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^{t,m} W^m h_j^t \right) \right. \quad (4.10)$$

where M denotes the number of attention heads.

The structural attention mechanism enables the model to focus on influential communication relationships associated with anomalous behavior.

4.1.5 Temporal Self-Attention Module

While structural attention captures spatial dependencies within each graph snapshot, temporal self-attention models topology evolution and interaction drift across different time windows.

For node i , the structural embeddings across all timestamps are represented as:

$$H_i = \{h_i^{1,struct}, h_i^{2,struct}, \dots, h_i^{T,struct}\} \quad (4.11)$$

Temporal attention computes the relevance between different timestamps using query-key similarity:

$$\beta_{t\tau} = \frac{\exp(q_t^T k_\tau)}{\sum_{\kappa=1}^T \exp(q_t^T k_\kappa)} \quad (4.12)$$

where:

- q_t is the query vector.
- k_τ is the key vector.
- v_τ is the value vector.

The temporally refined embedding is then computed as:

$$h_i^{t,temp} = \sum_{\tau=1}^T \beta_{t\tau} v_\tau \quad (4.13)$$

Temporal self-attention allows the model to capture long-range temporal dependencies and gradual anomaly emergence without relying solely on recurrent updates.

4.1.6 LSTM-Based Temporal Refinement

Although temporal attention effectively captures topology evolution, persistent anomalous behavior may span longer temporal durations. Therefore, DySTAL-AD integrates an LSTM refinement module to model long-term behavioral dependencies.

Let z_t denote the temporally refined embedding at time t . The LSTM performs sequential updates using gating mechanisms.

The forget gate is defined as:

$$f_t = \sigma(W_f z_t + U_f h_{t-1} + b_f) \quad (4.14)$$

The input gate is:

$$i_t = \sigma(W_i z_t + U_i h_{t-1} + b_i) \quad (4.15)$$

The candidate cell state is:

$$\tilde{c}_t = \tanh(W_c z_t + U_c h_{t-1} + b_c) \quad (4.16)$$

The memory cell update becomes:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (4.17)$$

The output gate is:

$$o_t = \sigma(W_o z_t + U_o h_{t-1} + b_o) \quad (4.18)$$

Finally, the hidden state is updated as:

$$h_t = o_t \odot \tanh(c_t) \quad (4.19)$$

The final hidden state h_T represents long-term spatio-temporal behavior.

4.1.7 Binary Anomaly Classification

The final hidden representation is mapped into anomaly probability using a fully connected classifier followed by sigmoid activation:

$$\hat{y} = \sigma(W_c h_T + b_c) \quad (4.20)$$

where:

- $\hat{y} \in [0, 1]$ represents anomaly probability.

- W_c and b_c are learnable classifier parameters.

4.1.8 Loss Function

Since anomaly detection datasets are highly imbalanced, weighted binary cross-entropy loss is employed:

$$L = -w_1 y \log(\hat{y}) - w_0 (1 - y) \log(1 - \hat{y}) \quad (4.21)$$

where:

- $y \in \{0, 1\}$ is the ground-truth label.
- w_0 and w_1 are class weights.

The weighted loss improves sensitivity toward minority anomalous samples.

4.2 CaT-GNN [1]: Causal-Temporal Graph Neural Network

4.2.1 Overview

CaT-GNN [1] is for reliable and explainable fraud detection in dynamic financial transaction systems [1]. The model integrates temporal graph learning with causal inference to distinguish causal transaction patterns from spurious correlations.

The architecture consists of:

1. Temporal Graph Encoder
2. Causal Inspector Module
3. Causal Intervener Module
4. Fraud Classification Layer

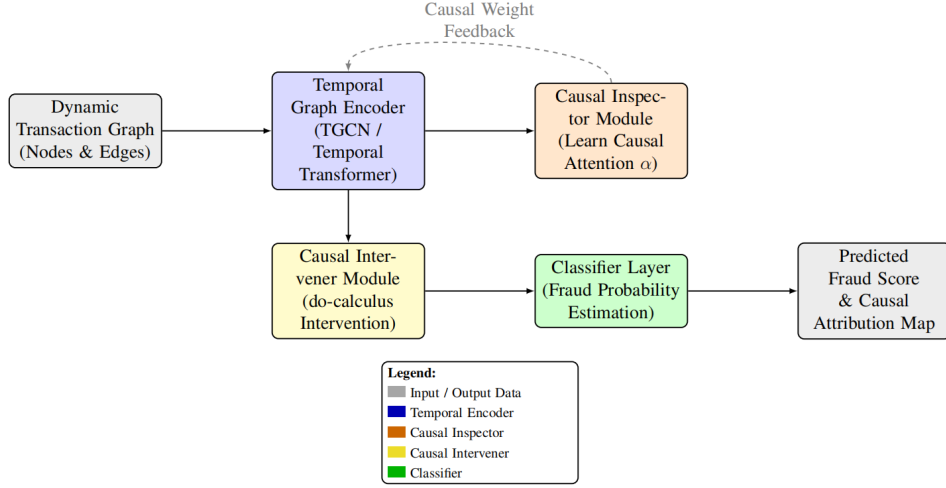


Figure 4.2: Architecture of the Proposed CaT-GNN model

Figure 4.2 presents the architecture of the proposed CaT-GNN [1] model for fraud detection in dynamic transaction systems. The model consists of a Temporal Graph Encoder to learn evolving transaction representation, a Causal Inspector Module [1] to learn causal attention weights, and a Causal Intervener Module [1] to perform intervention-based reasoning. The last classifier estimates the probability of fraud and creates causal attribution maps for better interpretability [1].

4.2.2 Dynamic Transaction Graph Representation

The financial system is represented as a temporal graph sequence:

$$\{G_1, G_2, \dots, G_T\} \quad (4.22)$$

where:

$$G_t = (V_t, E_t, X_t) \quad (4.23)$$

Here:

- V_t represents accounts or users.
- E_t represents financial transactions.
- X_t denotes node feature matrices.

The objective is to predict fraud probability:

$$\hat{y}_i^t = f_\theta(G_{\leq t}) \quad (4.24)$$

4.2.3 Temporal Graph Encoder

The Temporal Graph Encoder learns evolving transaction embeddings [1].

The embedding update is computed as:

$$h_i^t = \sigma \left(W_1 X_i^t + W_2 \sum_{j \in N(i)} \frac{1}{|N(i)|} h_j^{t-1} + b \right) \quad (4.25)$$

where:

- $N(i)$ denotes neighboring transactions.
- W_1 and W_2 are learnable parameters.

This operation captures both temporal continuity and relational dependencies.

4.2.4 Causal Inspector Module

The Causal Inspector identifies causally influential relationships using attention-based causal reasoning.

The causal attention weight is computed as:

$$\alpha_{ij}^t = \text{softmax}_j(\phi(h_i^t, h_j^t)) \quad (4.26)$$

where $\phi(\cdot)$ is implemented using a multilayer perceptron.

The causal embedding becomes:

$$c_i^t = \sum_{j \in N(i)} \alpha_{ij}^t h_j^t \quad (4.27)$$

This suppresses noisy correlations while emphasizing causally important interactions.

4.2.5 Causal Intervener Module

The Causal Intervener improves robustness through intervention-based reasoning inspired by Pearl’s do-calculus.

The intervened embedding is:

$$\tilde{h}_i^t = h_i^t + \delta_i^t \quad (4.28)$$

where:

$$\delta_i^t = g_{cf}(h_i^t, X_{\neg i}^t) \quad (4.29)$$

The intervention simulates counterfactual scenarios by perturbing non-causal features.

The causal regularization loss is defined as:

$$L_{causal} = ||\hat{y}_i^t - \hat{y}_i^{t,do}||_2^2 \quad (4.30)$$

This forces prediction consistency under interventions.

4.2.6 Fraud Classification

The final representation is formed by concatenating multiple embeddings:

$$z_i^t = [h_i^t || c_i^t || \tilde{h}_i^t] \quad (4.31)$$

The fraud probability is computed as:

$$\hat{y}_i^t = \sigma(Wz_i^t + b) \quad (4.32)$$

4.2.7 Overall Training Objective

The overall loss combines supervised classification and causal regularization:

$$L = L_{sup} + \lambda_c L_{causal} \quad (4.33)$$

where:

$$L_{sup} = - \sum_i [y_i^t \log(\hat{y}_i^t) + (1 - y_i^t) \log(1 - \hat{y}_i^t)] \quad (4.34)$$

Here:

- L_{sup} is supervised binary cross-entropy loss.
- L_{causal} improves causal robustness.
- λ_c controls causal regularization strength.

The model parameters are optimized in an end-to-end fashion using the Adam optimizer with learning rate scheduling and early stopping [\[1\]](#).

4.3 STARGATE: Spatio-Temporal Anomaly Recognition with Graph Attention Networks and Transformer Encoder

4.3.1 Overview

STARGATE is a hybrid Graph Attention Network (GAT) and Transformer-based model designed for anomaly detection in satellite telemetry systems. The model jointly models spatial dependencies among telemetry channels and long-range temporal dependencies in telemetry sequences. STARGATE integrates graph-based spatial representation learning with Transformer-based temporal sequence modeling to effectively capture evolving telemetry behavior and anomalous system dynamics.

The overall architecture consists of the following stages:

1. Telemetry sequence construction
2. Graph-based spatial dependency modeling
3. Graph Attention Network (GAT) learning
4. Transformer-based temporal learning
5. Anomaly classification

The model converts telemetry observations into spatio-temporal graph representations and learns telemetry-aware temporal embeddings for robust anomaly detection.

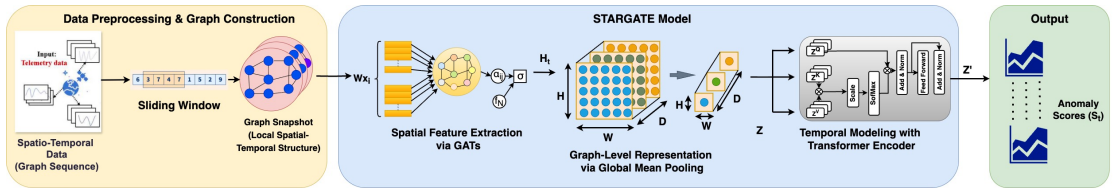


Figure 4.3: Architecture of the Proposed STARGATE model

Figure 4.3 illustrates the overall architecture of the proposed STARGATE model. The model first constructs telemetry graph sequences using temporal windows. A Graph Attention Network learns spatial dependencies among telemetry channels, while the Transformer encoder captures long-range temporal dependencies across telemetry sequences. The learned spatio-temporal representations are passed through fully connected layers for anomaly classification.

4.3.2 Problem Formulation

Let the telemetry sequence dataset be represented as:

$$X = \{x_1, x_2, x_3, \dots, x_N\} \quad (4.35)$$

where each telemetry observation x_i contains multivariate sensor measurements collected from spacecraft telemetry channels.

The telemetry data is segmented into temporal windows to construct graph sequences:

$$G = \{G_1, G_2, G_3, \dots, G_T\} \quad (4.36)$$

where each graph snapshot is represented as:

$$G_t = (V_t, E_t) \quad (4.37)$$

Here:

- V_t denotes telemetry channels.
- E_t represents correlations among telemetry channels.

The objective of the model is to learn a mapping function:

$$f : G \rightarrow \{0, 1\} \quad (4.38)$$

where:

- 0 represents normal telemetry behavior.
- 1 represents anomalous telemetry behavior.

4.3.3 Telemetry Sequence Construction

The telemetry data is segmented into fixed-length temporal windows using a sliding window mechanism. Each window captures sequential telemetry behavior over time.

For each temporal window, telemetry graphs are constructed where nodes correspond to telemetry channels and edges represent channel relationships computed using correlation-based similarity measures.

Each telemetry node is assigned a feature vector:

$$x_i^t = [f_1, f_2, f_3, \dots, f_d] \quad (4.39)$$

where the features may include:

- Sensor measurements
- Channel statistics
- Temporal variations
- Frequency information
- Telemetry behavioral features

The generated graph sequence:

$$G = \{G_1, G_2, \dots, G_T\} \quad (4.40)$$

captures evolving telemetry dependencies and temporal system behavior.

4.3.4 Graph Attention Network Module

The Graph Attention Network module captures spatial dependencies among telemetry channels within each graph snapshot. Instead of assigning equal importance to neighboring telemetry channels, the attention mechanism learns adaptive importance weights.

For telemetry node i and neighboring node j , the attention coefficient is computed as:

$$\alpha_{ij}^t = \frac{\exp(\text{LeakyReLU}(a^T [Wh_i^t || Wh_j^t]))}{\sum_{k \in N(i)} \exp(\text{LeakyReLU}(a^T [Wh_i^t || Wh_k^t]))} \quad (4.41)$$

where:

- W is the learnable weight matrix.
- a is the attention vector.
- $||$ denotes concatenation.
- $N(i)$ represents neighboring telemetry channels.

The node embedding update is computed as:

$$h_i^{t,spatial} = \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^t Wh_j^t \right) \quad (4.42)$$

Multi-head attention is further utilized to improve representation stability:

$$h_i^{t,spatial} = \left\|_{m=1}^M \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^{t,m} W^m h_j^t \right) \right. \quad (4.43)$$

where M denotes the number of attention heads.

The GAT module enables the model to focus on important telemetry relationships associated with anomalous system behavior.

4.3.5 Transformer-Based Temporal Learning

The Transformer encoder captures long-range temporal dependencies among telemetry sequences using self-attention mechanisms.

For the telemetry embedding sequence:

$$H = \{h_1, h_2, \dots, h_T\} \quad (4.44)$$

query, key, and value representations are computed as:

$$Q = HW_Q \quad (4.45)$$

$$K = HW_K \quad (4.46)$$

$$V = HW_V \quad (4.47)$$

where W_Q , W_K , and W_V are learnable projection matrices.

The self-attention operation is computed as:

$$Attention(Q, K, V) = softmax \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (4.48)$$

where d_k represents the dimensionality of the key vectors.

Multi-head attention is applied as:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W_O \quad (4.49)$$

where:

$$head_i = Attention(Q_i, K_i, V_i) \quad (4.50)$$

The Transformer encoder effectively captures long-range temporal dependencies and evolving telemetry behavior.

4.3.6 Anomaly Classification

The final Transformer representation is passed through fully connected layers followed by sigmoid activation for anomaly prediction:

$$\hat{y} = \sigma(W_ch_T + b_c) \quad (4.51)$$

where:

- $\hat{y} \in [0, 1]$ represents anomaly probability.
- W_c and b_c are learnable classifier parameters.

4.3.7 Loss Function

Binary cross-entropy loss is utilized for anomaly classification:

$$L = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (4.52)$$

where:

- $y \in \{0, 1\}$ is the ground-truth label.
- $\hat{y}$ is the predicted anomaly probability.

The loss function enables effective optimization for telemetry anomaly detection.

Chapter 5

EXPERIMENTAL RESULTS AND DISCUSSION

This chapter presents the experimental evaluation of the proposed DySTAL-AD, CaT-GNN [1], and STARGATE models on benchmark datasets from cybersecurity, financial fraud detection, and satellite telemetry domains. The experiments evaluate the effectiveness of the proposed models in terms of anomaly detection capability, robustness, temporal learning, dynamic graph representation, Transformer-based sequence modeling, and causal reasoning. The proposed methods are compared against several baseline models including recurrent architectures, static graph neural networks, dynamic graph learning approaches, and Transformer-based anomaly detection models. The performance is evaluated using standard classification metrics including Accuracy, Precision, Recall, F1-score, and AUC-ROC.

5.1 Experimental Setup

All experiments are implemented using Python, PyTorch, and PyTorch Geometric models. The models are trained on GPU-enabled systems to accelerate dynamic graph computation and deep learning optimization. The structural self-attention and Graph Attention Network modules utilize four attention heads with an embedding dimension of 128. The LSTM refinement module in DySTAL-AD consists of two stacked recurrent layers for modeling long-term temporal dependencies, while the STARGATE model employs Transformer encoder layers for long-range telemetry sequence learning. A dropout rate of 0.5 is applied to reduce overfitting and improve model generalization capability.

The models are optimized using the AdamW optimizer with a learning rate of 10^{-3} . Training is performed for a maximum of 100 epochs with early stopping

based on validation F1-score to prevent overfitting. Gradient clipping is additionally employed to stabilize recurrent and Transformer-based optimization. The datasets are divided into training, validation, and testing subsets using a stratified 70:15:15 split to preserve anomaly class distribution across all subsets.

5.2 Evaluation Metrics

The proposed models are evaluated using multiple classification metrics to provide comprehensive analysis of anomaly detection performance.

5.2.1 Accuracy

Accuracy measures the proportion of correctly classified samples and is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives respectively.

5.2.2 Precision

Precision measures the correctness of positive anomaly predictions:

$$Precision = \frac{TP}{TP + FP}$$

Higher precision indicates lower false positive rates.

5.2.3 Recall

Recall measures the capability of the model to correctly identify anomalous samples:

$$Recall = \frac{TP}{TP + FN}$$

Higher recall indicates better anomaly detection sensitivity.

5.2.4 F1-score

F1-score balances precision and recall:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

It is particularly important for highly imbalanced anomaly detection datasets.

5.2.5 AUC-ROC

Area Under the Receiver Operating Characteristic Curve (AUC-ROC) measures the separability between normal and anomalous classes across multiple thresholds. Higher AUC values indicate stronger classification capability and better discrimination between normal and anomalous samples.

5.3 Results of DySTAL-AD Model

The DySTAL-AD model is evaluated on CICIDS-2017, CICIDS-2018, and CICIoT-2022 datasets. The proposed model is compared against baseline models including GCN, GAT, GRU, and DySAT.

5.3.1 Results on CICIDS-2017 Dataset

Table 5.1 presents the performance results on the CICIDS-2017 dataset.

Table 5.1: Performance results on CICIDS-2017

Model	Precision	Recall	F1-score
GCN	0.9843	0.9872	0.9858
GAT	0.9920	0.9902	0.9908
GRU	0.9923	0.9872	0.9898
DySAT	0.9900	0.9964	0.9924
DySTAL-AD	0.9944	0.9982	0.9958

The results indicate that DySTAL-AD achieves the best overall performance across all baseline models. The model achieved the highest Precision, Recall and F1-score, which indicates its better ability to detect anomalous network behavior. The improved recall value shows that the combination of structural self-attention, temporal self-attention and recurrent refinement can detect persistent anomalous communication patterns. Compared with DySAT, the extra LSTM refinement layer enhances temporal consistency and long-term anomaly modeling.

5.3.2 Results on CICIDS-2018 Dataset

Table 5.2 presents the performance comparison on the CICIDS-2018 dataset.

Table 5.2: Performance Comparison on CICIDS-2018

Model	Precision	Recall	F1-score
GCN	0.8377	0.9209	0.8687
GAT	0.8410	0.9121	0.8700
GRU	0.8325	0.9167	0.8680
DySAT	0.8400	0.9156	0.8703
DySTAL-AD	0.8467	0.9279	0.8787

Due to the larger scale and more diverse attack scenarios, the anomaly detection of CICIDS-2018 dataset is much more difficult. DySTAL-AD obtains consistently better results than all the baselines in all evaluation metrics. The improvement in F1-score shows better balance between precision and recall in highly dynamic network environments. Results indicate that the proposed model is effective in capturing the changing communication behavior and topology drift of large scale enterprise networks.

5.3.3 Results on CICIoT-2022 Dataset

Table 5.3 presents the performance comparison on the CICIoT-2022 dataset.

Table 5.3: Performance Comparison on CICIoT-2022

Model	Precision	Recall	F1-score
GCN	0.7472	0.7942	0.7625
GAT	0.7343	0.7693	0.7245
GRU	0.8116	0.8102	0.8123
DySAT	0.7786	0.7995	0.7802
DySTAL-AD	0.8186	0.8162	0.8109

The CICIoT-2022 dataset reflects the highly heterogeneous IoT communication environments with evolving device interactions and dynamic topology changes. DySTAL-AD shows the highest precision and recall values, demonstrating the strong generalization capability under complex IoT conditions. Although the GRU is slightly competitive with F1-score, the proposed model offers better structural-temporal integration by means of dynamic graph learning and attention mechanisms. The results show the robustness of DySTAL-AD for real world IoT anomaly detection scenarios.

5.3.4 Accuracy Comparison Across Datasets

Figure 5.1 presents the accuracy comparison of different baseline models on the CICIDS-2017, CICIDS-2018, and CICIoT-2022 datasets. The proposed DySTAL-AD model consistently achieves the highest accuracy across all datasets, demonstrating its effectiveness in modeling evolving communication patterns and dynamic graph structures.

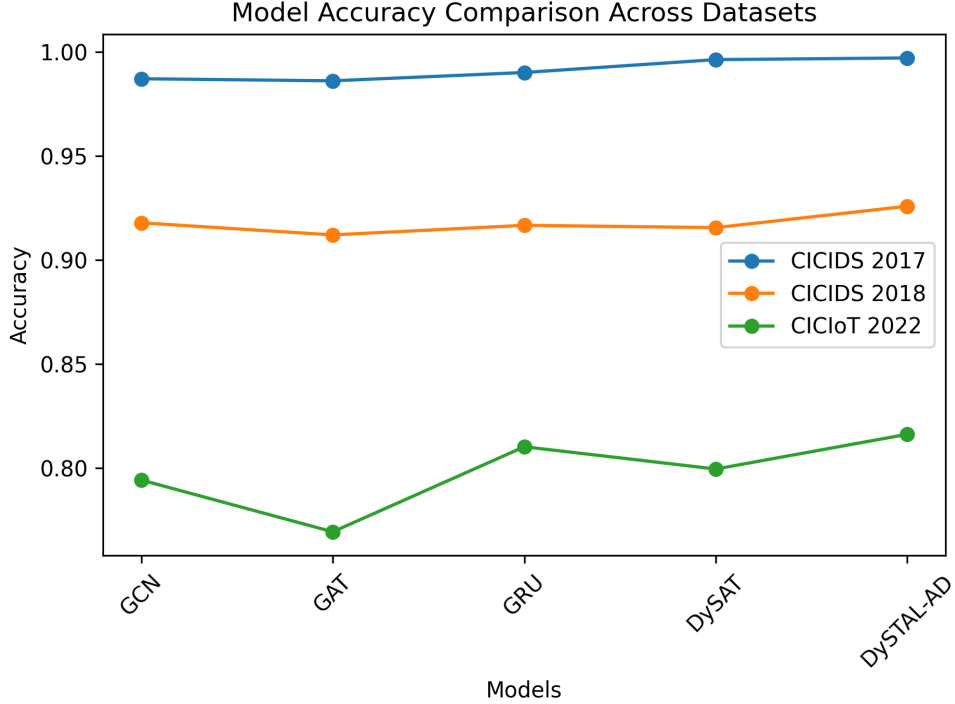


Figure 5.1: Accuracy Comparison of Different Models Across CICIDS-2017, CICIDS-2018, and CICIoT-2022 Datasets

The results demonstrate that conventional graph neural networks such as GCN and GAT can achieve satisfactory performance by modeling structural dependencies, but they are not efficient to model the temporal evolution and the topology drift. GRU and other recurrent models, enhance the temporal learning capability. However, they do not well model the relational graph interactions. Dynamic graph learning methods (e.g., DySAT) can combine the structure and temporal attention mechanism to achieve better performance. However, the proposed DySTAL-AD model is better in accuracy because it integrates dynamic graph construction, structural self-attention, temporal self-attention, and LSTM-based temporal refinement together.

In the experiments, DySTAL-AD performs best on CICIDS-2017 dataset, reaching an accuracy of 99.7% or above, demonstrating a good intrusion detection ability in the enterprise network environment. Similarly, on the CICIDS-2018 dataset,

the model obtains the best performance with the accuracy of about 92.6%, which demonstrates the robustness of the model under more complex and large-scale attack scenarios. DySTAL-AD also achieves superior performance over baseline models on the CICIoT-2022 dataset by effectively addressing the heterogeneous IoT communication behavior and dynamic topology evolution.

In summary, the results verify that the combination of dynamic graph learning with attention mechanism and recurrent refinement significantly enhances the anomaly detection performance in the dynamic network systems.

5.4 Results of CaT-GNN model

The CaT-GNN [1] model is evaluated on the Elliptic Bitcoin Transaction Dataset and PaySim Mobile Money Dataset for fraud detection.

5.4.1 AUC Performance Comparison

Figure 5.2 shows the AUC comparison of different baseline models on the Elliptic and PaySim datasets. The proposed CaT-GNN [1] model achieves the highest AUC score of 0.757 for Elliptic and 0.864 for PaySim dataset, outperforming the recurrent, static graph and temporal graph learning baselines. This validation shows that the combination of the causal reasoning and the temporal graph representation learning is effective for robust fraud detection under the evolving transaction behavior.

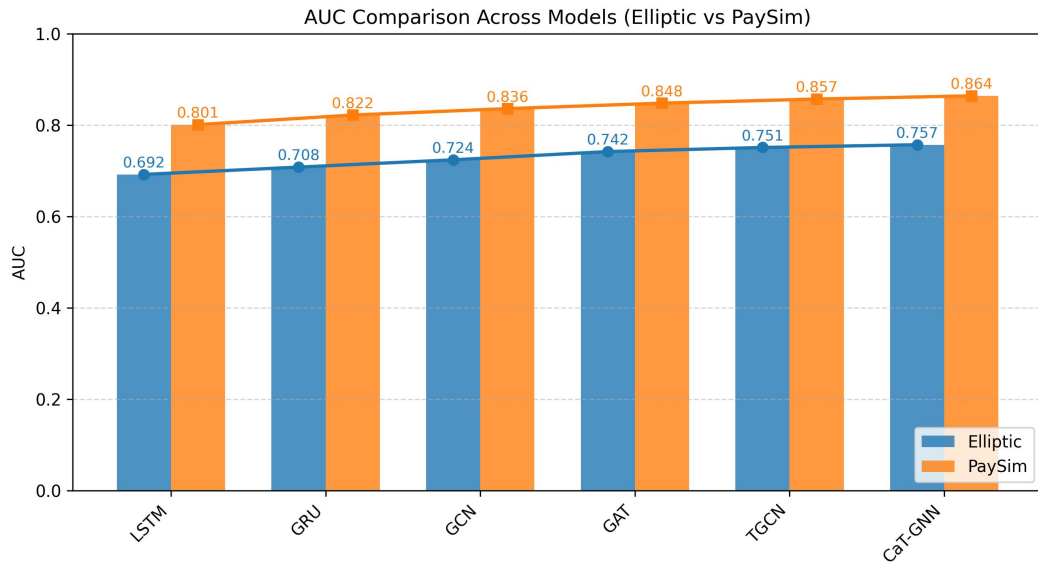


Figure 5.2: Cross Model AUC Comparison for PaySim and Elliptic Datasets

5.4.2 Precision Performance Comparison

The comparison of the precision across different models is illustrated in Figure 5.3. CaT-GNN [1] obtains highest precision values on both datasets, showing it can reduce false positive predictions while maintaining strong fraud detection performance. The causal intervention mechanism directs the model to pay attention to transaction relationships that are causally important, rather than noisy correlations.

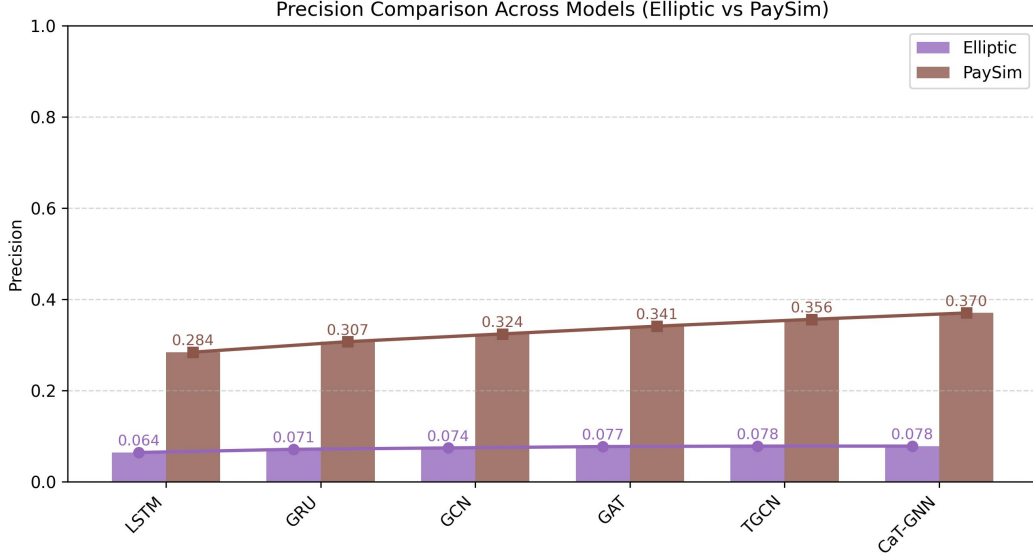


Figure 5.3: Cross Model Precision Comparison for Elliptic and PaySim Datasets

5.4.3 F1-score Performance Comparison

Figure 5.4 shows the comparison of F1 score between baseline models. The proposed CaT-GNN [1] consistently obtains the best F1-score on both datasets, which indicates the improvement of the balance between precision and recall. The results confirm that the combination of temporal graph learning and causal reasoning enhances robustness and generalization ability in the dynamic financial transaction settings.

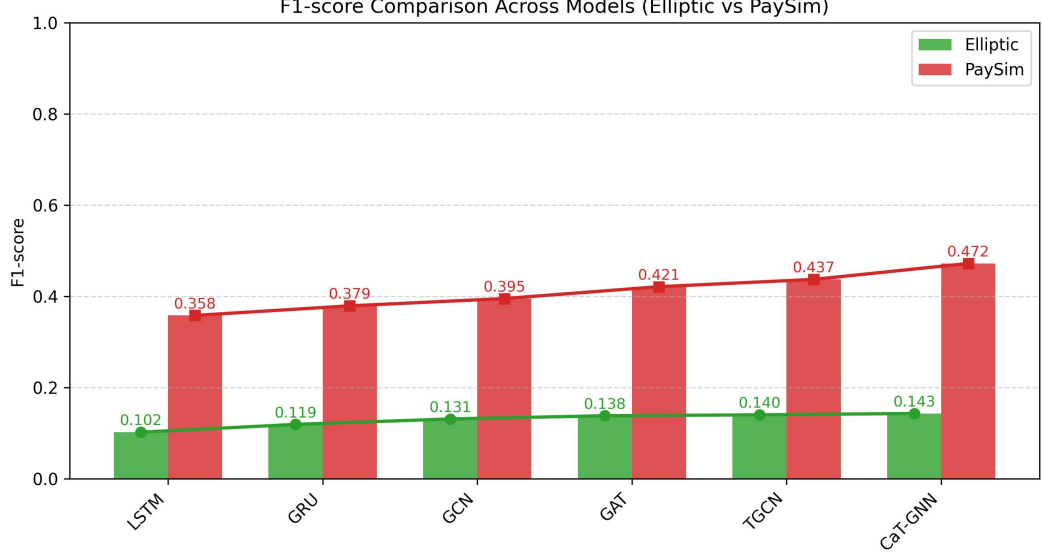


Figure 5.4: Cross Model F1-Score Comparison for Elliptic and PaySim Datasets

5.4.4 Discussion

Experimental results demonstrate that the proposed CaT-GNN [1] outperforms the traditional recurrent models, static graph neural networks and temporal graph learning baselines in all evaluation metrics. LSTM and GRU can capture temporal dependencies but they cannot model relational transaction structures, while GCN and GAT can capture graph dependencies but they do not model temporal evolution well. Temporal graph models such as TGCN enhance dynamic representation learning but are susceptible to false correlations and noisy transaction behaviors. To overcome these limitations, we propose a CaT-GNN [1] model that integrates temporal graph learning with causal reasoning and intervention-based regularization. Dynamic transaction interactions are learned by the Temporal Graph Encoder. Causal relationships are detected by the Causal Inspector, which filters out noisy correlations. Moreover, the Causal Intervener increases the robustness to temporal drift and changing transaction distributions. Thus, CaT-GNN [1] obtains better fraud detection accuracy, robustness, interpretability and generalization ability in both cryptocurrency and mobile financial transaction datasets.

5.5 Experimental Results of STARGATE

This section presents the experimental evaluation of the proposed STARGATE framework on satellite telemetry anomaly detection datasets. The framework is compared against several baseline anomaly detection models including LSTM, One-Class SVM (OCSVM), OSHAD-CSL, Variational Graph Encoder (VGE), and

TA-LSTM-AE. The experiments analyze the effectiveness of STARGATE in terms of Precision, Recall, F1-score, convergence behavior, and training stability under different architectural configurations.

5.5.1 Comparative Performance Analysis

The comparative performance evaluation demonstrates that the proposed STARGATE framework achieves superior anomaly detection capability compared to existing baseline approaches. The integration of Graph Attention Networks and Transformer encoders enables the framework to jointly model spatial telemetry dependencies and long-range temporal behavior more effectively than traditional recurrent and graph-based methods.

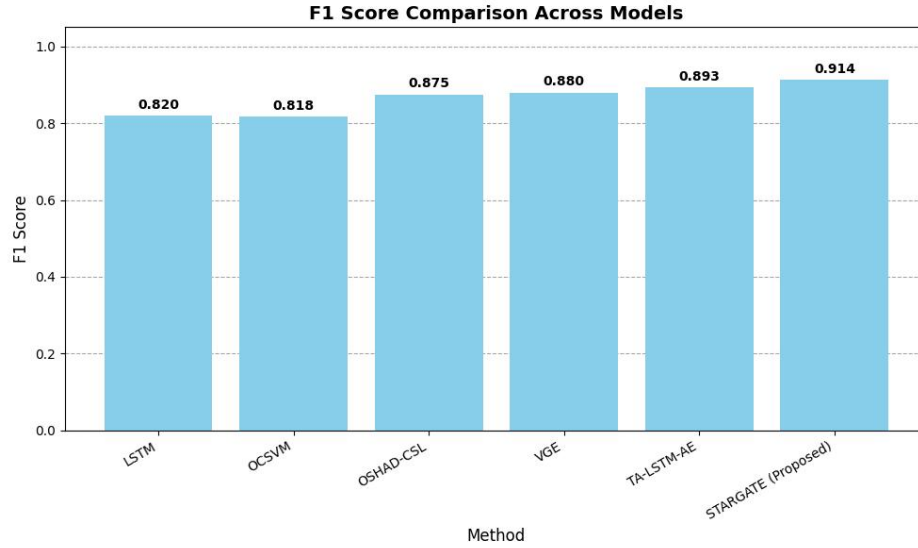


Figure 5.5: F1-score Comparison Across Different Models

Figure 5.5 shows the F1-score comparison across different anomaly detection models. The proposed STARGATE framework achieves the highest F1-score of 0.914, outperforming LSTM, OCSVM, OSHAD-CSL, VGE, and TA-LSTM-AE. The improvement demonstrates the effectiveness of combining graph-based spatial learning with Transformer-based temporal modeling for telemetry anomaly detection.

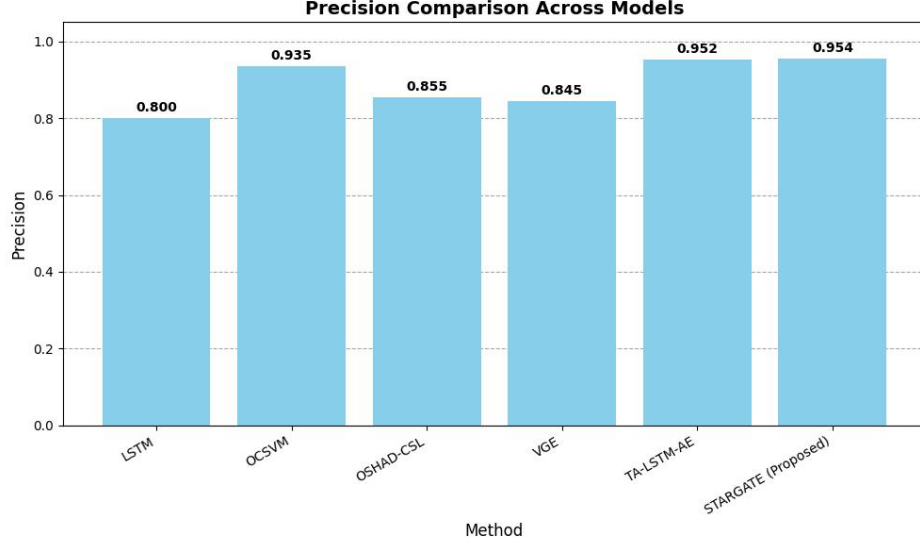


Figure 5.6: Precision Comparison Across Different Models

Figure 5.6 presents the precision comparison of different models. STARGATE achieves the highest precision value of 0.954, indicating lower false positive rates and improved anomaly discrimination capability compared to baseline methods.

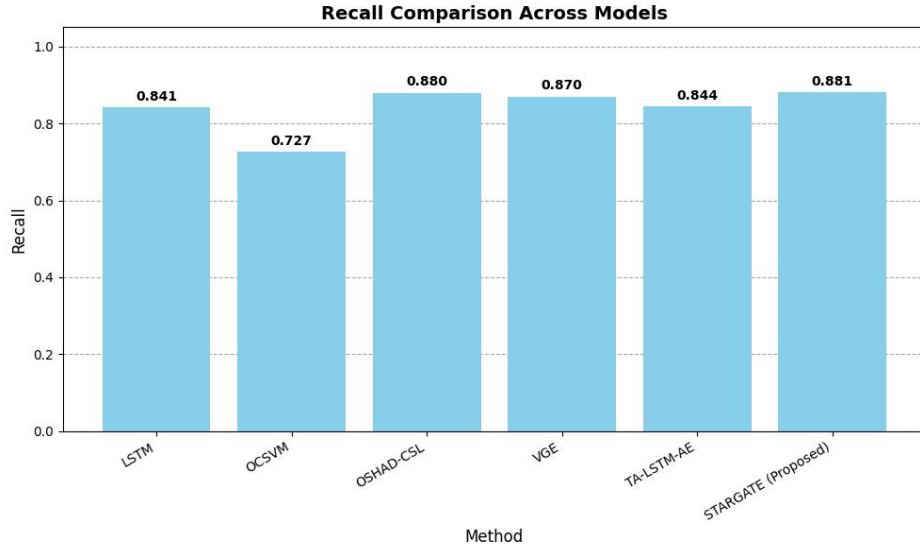


Figure 5.7: Recall Comparison Across Different Models

Figure 5.7 illustrates the recall performance comparison across models. STARGATE achieves a recall value of 0.881, demonstrating strong capability in correctly identifying anomalous telemetry behavior while maintaining stable prediction performance.

5.5.2 Effect of Window Length

The effect of temporal window length on model performance and computational complexity is analyzed to determine the optimal sequence length for telemetry anomaly detection.

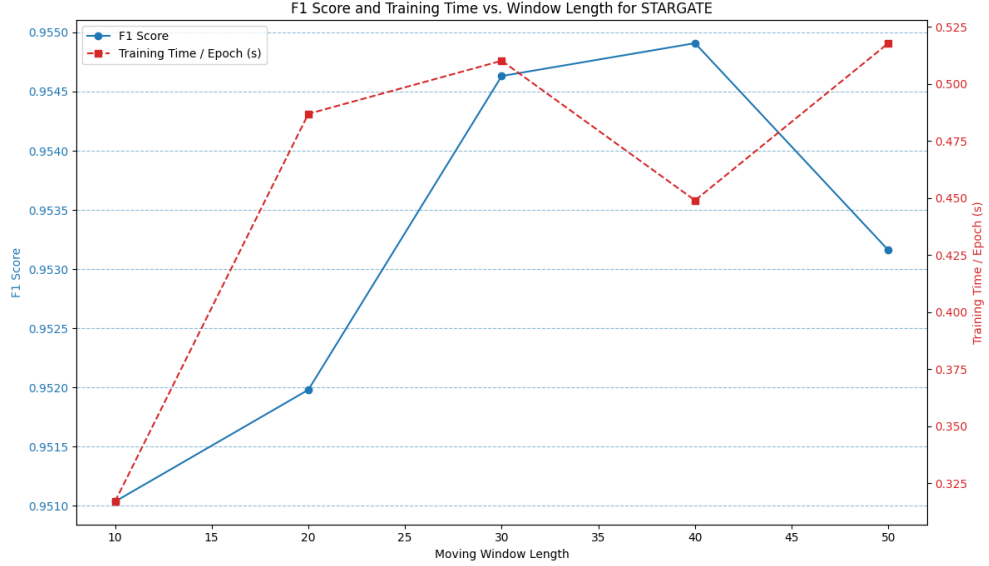


Figure 5.8: F1-score and Training Time versus Window Length

Figure 5.8 shows the relationship between F1-score and training time for different temporal window lengths. The results indicate that moderate window lengths achieve better anomaly detection performance while maintaining computational efficiency. Larger window sizes slightly improve temporal dependency modeling but increase training complexity.

5.5.3 Training Loss Analysis for Hidden Nodes

The influence of hidden node dimensions on training convergence is analyzed to evaluate model stability and learning behavior.

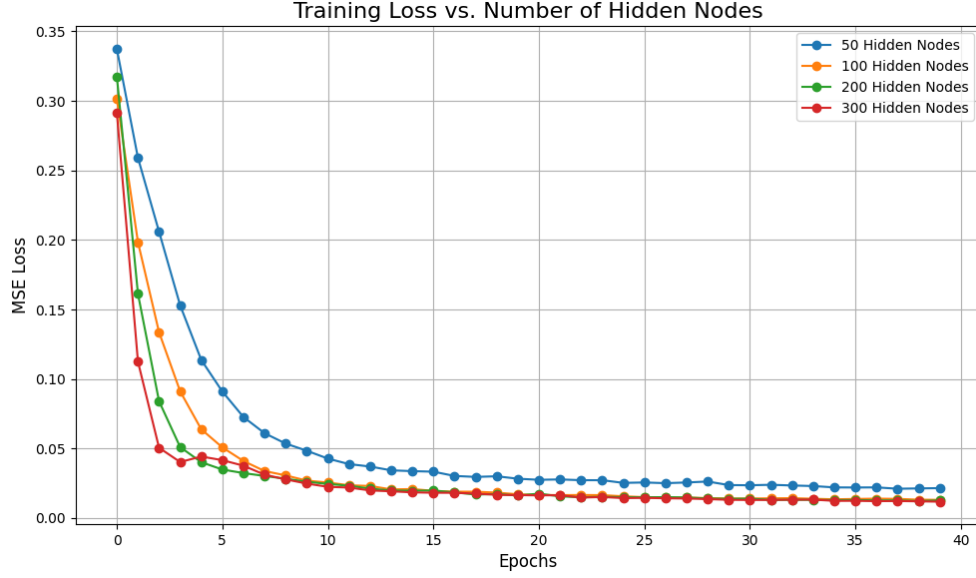


Figure 5.9: Training Loss versus Number of Hidden Nodes

Figure 5.9 presents the training loss curves for different hidden node configurations. The results show that increasing hidden dimensions improves convergence speed and representation capability up to an optimal threshold, after which the performance gain becomes marginal.

5.5.4 Training Loss Analysis for Transformer Layers

The effect of Transformer encoder depth on training performance is further evaluated using different numbers of Transformer layers.

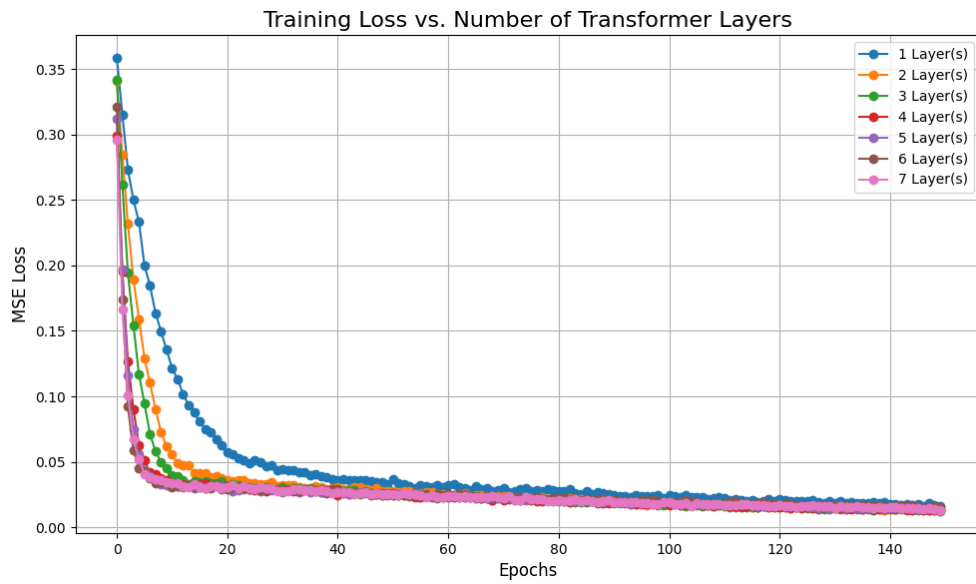


Figure 5.10: Training Loss versus Number of Transformer Layers

Figure 5.10 shows the training loss behavior for varying Transformer depths. The results indicate that deeper Transformer architectures improve temporal representation learning and convergence stability, although excessive depth may increase computational complexity.

5.5.5 Validation Loss Analysis for Hidden Nodes

Validation performance under different hidden node configurations is analyzed to evaluate generalization capability.

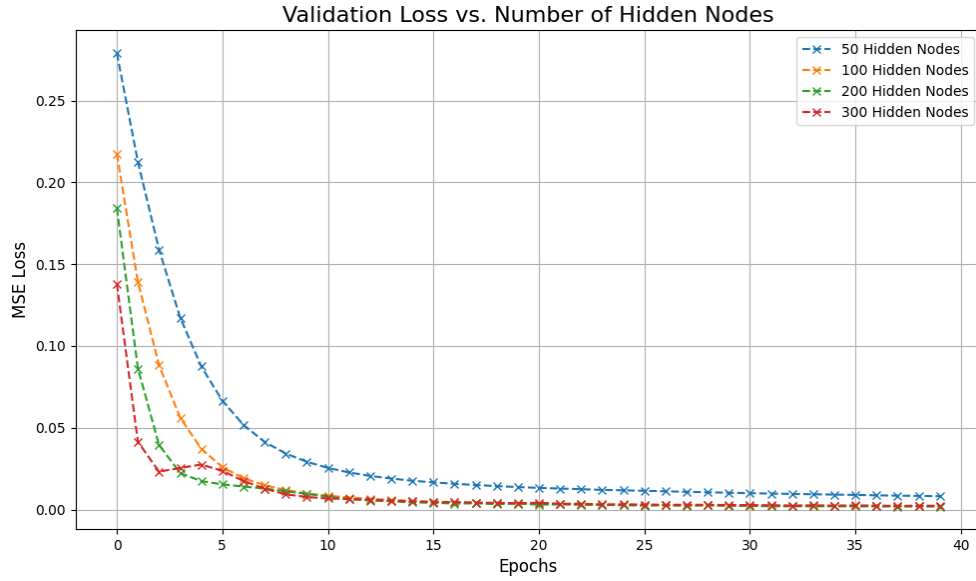


Figure 5.11: Validation Loss versus Number of Hidden Nodes

Figure 5.11 illustrates the validation loss curves for different hidden node dimensions. The results demonstrate stable generalization performance with reduced validation loss for moderate hidden node configurations.

5.5.6 Validation Loss Analysis for Transformer Layers

The effect of Transformer depth on validation performance is evaluated using multiple encoder layer configurations.

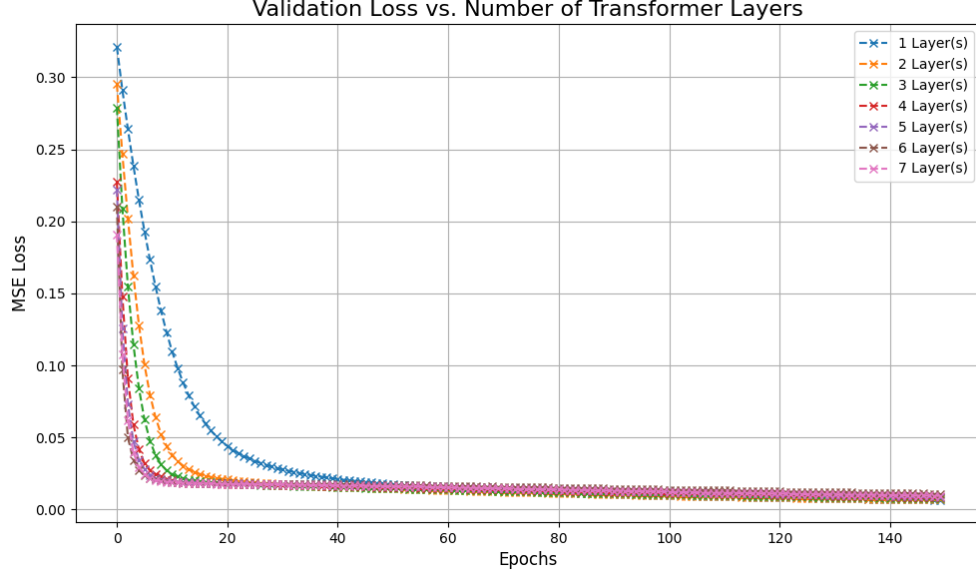


Figure 5.12: Validation Loss versus Number of Transformer Layers

Figure 5.12 presents the validation loss comparison for different Transformer layer configurations. The results show that multi-layer Transformer encoders improve temporal learning capability and anomaly detection robustness while maintaining stable generalization performance.

5.5.7 Discussion

The experimental results demonstrate that the proposed STARGATE framework effectively captures both spatial telemetry relationships and long-range temporal dependencies through the integration of Graph Attention Networks and Transformer encoders. Compared to conventional recurrent and graph-based methods, STARGATE achieves superior Precision, Recall, and F1-score performance while maintaining stable convergence and strong generalization capability. The analysis of hidden node dimensions, Transformer depth, and temporal window size further confirms the robustness and scalability of the proposed framework for satellite telemetry anomaly detection.

5.6 Comparative Discussion

The overall experimental results demonstrate that the proposed DySTAL-AD, CaT-GNN [1], and STARGATE frameworks consistently outperform or achieve highly competitive performance compared to existing baseline methods across cybersecurity, financial fraud detection, and satellite telemetry datasets.

The superior performance of DySTAL-AD is primarily attributed to the effective integration of dynamic graph construction, structural self-attention, temporal self-attention, and recurrent memory refinement, which jointly enable the framework to capture evolving communication behavior, topology drift, and long-term anomaly persistence more effectively than static graph or temporal-only approaches. Similarly, the improved performance of CaT-GNN [1] demonstrates the importance of incorporating causal reasoning into dynamic graph learning. The Causal Inspector and Causal Intervener modules improve robustness and interpretability by suppressing spurious correlations and enforcing prediction consistency under intervention-based perturbations. The STARGATE framework further demonstrates that combining Graph Attention Networks with Transformer encoders significantly improves telemetry anomaly detection by jointly modeling spatial telemetry dependencies and long-range temporal behavior.

The experiments further establish that dynamic graph learning substantially improves anomaly detection capability, while structural and temporal attention mechanisms enhance representation quality and long-term behavioral modeling. Transformer-based temporal learning improves sequential dependency capture in telemetry systems, and causal reasoning enhances robustness and interpretability under evolving conditions. Overall, the proposed frameworks demonstrate strong generalization capability across cybersecurity, financial, and satellite telemetry domains, establishing that integrating graph representation learning, temporal modeling, self-attention mechanisms, recurrent refinement, Transformer architectures, and causal reasoning provides a robust and scalable solution for anomaly detection in evolving complex network systems..

Chapter 6

CONCLUSION AND FUTURE WORK

In this thesis, we proposed three advanced Graph Neural Network based spatio-temporal frameworks, namely DySTAL-AD, CaT-GNN [1], and STARGATE, for anomaly detection in dynamic IoT communication networks, financial transaction systems, and satellite telemetry environments. DySTAL-AD integrated dynamic graph construction, structural self-attention, temporal self-attention, and LSTM refinement to capture evolving communication behavior and long-term anomalies. CaT-GNN [1] combined temporal graph learning with causal reasoning and intervention-based regularization to improve robustness, interpretability, and fraud detection capability under temporal drift and evolving transaction patterns. STARGATE further integrated Graph Attention Networks and Transformer encoders to jointly model spatial telemetry dependencies and long-range temporal behavior for satellite telemetry anomaly detection. Experimental results on benchmark datasets including CICIDS-2017, CICIDS-2018, CICIOT-2022, Elliptic, PaySim, NASA SMAP, and NASA MSL demonstrated that the proposed frameworks achieved superior or highly competitive performance compared to existing baseline methods in terms of Accuracy, Precision, Recall, F1-score, and AUC-ROC. Future work may focus on real-time streaming anomaly detection, lightweight edge-deployable graph neural networks, self-supervised graph representation learning, explainable AI techniques, federated graph learning, and more efficient Transformer-based temporal graph architectures for scalable, adaptive, and privacy-preserving anomaly detection in dynamic complex network environments.

Chapter 6

BIBLIOGRAPHY

- [1] D. Mandal and S. Kumar, “Cat-gnn: A causal-temporal graph neural network for robust and interpretable fraud detection in dynamic transaction networks,” in *Intelligent Strategies for ICT*. Cham: Springer Nature Switzerland, 2026, pp. 370–381.
- [2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proceedings of ICISSP*, 2018.
- [3] A. H. Lashkari, Y. Zang, G. Owhuo, and M. Mamun, “Characterization of cids2018 dataset,” in *Proceedings of ICISSP*, 2019.
- [4] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Telemetry anomaly detection for space missions using recurrent neural networks,” *arXiv preprint arXiv:1802.04431*, 2018.
- [5] —, “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding,” *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 387–395, 2018.
- [6] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
- [7] R. Vinayakumar, K. Soman, and P. Poornachandran, “Deep learning approach for intelligent intrusion detection system,” *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.
- [8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- [9] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [10] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [12] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [13] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, C. Leiserson, and H. Chen, “Evolvegc: Evolving graph convolutional networks for dynamic graphs,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, 2020, pp. 5363–5370.
- [14] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, “Temporal graph networks for deep learning on dynamic graphs,” *arXiv preprint arXiv:2006.10637*, 2021.
- [15] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, “Dynamic graph representation learning via self-attention networks,” in *Proceedings of the 13th ACM International Conference on Web Search and Data Mining*, 2020, pp. 519–527.
- [16] F. Manessi, A. Rozza, and M. Manzo, “Dynamic graph convolutional networks,” *Pattern Recognition*, vol. 97, p. 107000, 2020.
- [17] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, “Inductive representation learning on temporal graphs,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [19] J. Yu, J. Xia, and H. Yin, “Dyformer: Dynamic graph transformer for temporal graph learning,” in *Proceedings of AAAI*, 2023.

- [20] J. Xu, H. Wu, J. Wang, and M. Long, “Anomaly transformer: Time series anomaly detection with association discrepancy,” in *International Conference on Learning Representations (ICLR)*, 2022.
- [21] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant risk minimization,” *arXiv preprint arXiv:1907.02893*, 2019.
- [22] B. Schölkopf, F. Locatello, S. Bauer, N. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, “Toward causal representation learning,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.
- [23] Y. Wang, Y. Cao, Y. Li, and H. Zhu, “Causalgnn: Causal-based graph neural networks for spatio-temporal data,” in *Proceedings of KDD*, 2021.
- [24] Z. Zhang, H. Liu, and J. Tang, “Causal temporal graph neural networks for dynamic graph learning,” in *Proceedings of ICDM*, 2022.
- [25] M. Weber, J. Domeniconi, G. Chen, J. Weidele, C. Bellei, T. Robinson, and C. Leiserson, “Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics,” in *KDD Workshop on Anomaly Detection in Finance*, 2019.
- [26] S. Rezaei and X. Liu, “Deep learning-based intrusion detection system for iot networks,” *IEEE Communications Letters*, vol. 23, no. 7, pp. 1248–1251, 2019.
- [27] Y. Zhang, X. Wang, and C. Shi, “Graph attention transformer networks for spatio-temporal learning,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [28] A. Suo, Y. Zhao, and H. Xu, “Graph neural networks for spacecraft telemetry anomaly detection,” *IEEE Aerospace Conference*, 2022.
- [29] X. Zheng, B. Aragam, P. Ravikumar, and E. Xing, “Dags with no tears: Continuous optimization for structure learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.