Leveraging Graph Attention and Temporal Fusion for Lane-Level Traffic Anomaly Detection in Vehicle-Road Collaborative Systems

by

Reaz Raza
22101632
Abrar Syed Chowdhury
22101571
Tanvir
22101337

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
October 2025

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Reaz Raza
22101632

Abrar Syed Chowdhury
22101571

Tanvir
22101337

Approval

The thesis titled "Leveraging Graph Attention and Temporal Fusion for Lane-Level Traffic Anomaly Detection in Vehicle-Road Collaborative Systems" submitted by

1. Reaz Raza (22101632)
2. Abrar Syed Chowdhury (22101571)
3. Tanvir (22101337)

of Summer, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 8, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty

Professor
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

As urban traffic systems grow increasingly complex, traditional methods for managing congestion and accidents are proving insufficient. This research proposes a lane-level traffic anomaly detection framework that integrates Graph Attention Networks (GAT) with temporal fusion encoding to capture spatio-temporal dependencies within vehicle-road collaborative systems. A self-supervised learning mechanism enables effective anomaly detection with minimal labeled data while an attention-based module prioritizes critical neighbour regions to enhance robustness and computational efficiency. Through comprehensive experimentation, the proposed framework demonstrates strong anomaly detection performance with an F1-score of 0.84, significantly outperforming established base models such as LSTM (0.74), Autoencoder (0.65), DBSCAN (0.46) etc. This enables more adaptive, efficient and intelligent traffic management while optimizing overall network flow.

Keywords: Graph Attention Networks, Temporal Fusion, Lane-Level Traffic Anomaly Detection, Vehicle-Road Collaborative Systems, Self-Supervised Learning, Contrastive Learning, Graph Neural Networks, Sensor Fusion, Time-Series Traffic Data, SUMO Simulation, OSM Road Network, Anomaly Scoring, Spatio-Temporal Modeling, Unsupervised Learning

Table of Contents

Declaration i

Approval ii

Abstract iii

Table of Contents iv

List of Figures vi

List of Tables vii

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Rationale of the Study . . . 1
- 1.3 Problem Statement . . . 2
- 1.4 Research Objective . . . 4
- 1.5 Methodology in Brief . . . 5
- 1.6 Scopes and Challenges . . . 6
  - 1.6.1 Scopes of the research . . . 6
  - 1.6.2 Challenges of the Research . . . 6

2 Literature Review 7
- 2.1 Preliminaries . . . 7
- 2.2 Review of Existing Research . . . 7
- 2.3 Summary of Key Findings . . . 15

3 Requirements, Impacts and Constraints 16
- 3.1 Final Specifications and Requirements . . . 16
- 3.2 Societal Impact . . . 17
- 3.3 Environmental Impact . . . 17
- 3.4 Ethical Issues . . . 18
- 3.5 Project Management Plan . . . 18
- 3.6 Risk Management . . . 19
- 3.7 Economic Analysis . . . 19

4 Proposed Methodology 20
- 4.1 Phases of the Methodology . . . 20
  - 4.1.1 Phase 1: Evaluating Traditional Models . . . 20
  - 4.1.2 Phase 2: Recognizing Research Limitations . . . 20

4.1.3 Phase 3: Adoption of a Graph-Based Framework 20
4.1.4 Phase 4: Designing Spatio-Temporal Graph Based Model for Anomaly Detection 21
4.2 Dataset Description 22
4.2.1 Simulation Setup 22
4.2.2 Feature Columns 23
4.2.3 Anomaly Injection 24
4.2.4 Statistical Summary of the Dataset 26
4.3 In-Built Models 27
4.4 Proposed Model Specification 31
4.4.1 Architecture Overview 32
4.4.2 Graph Construction and Feature Representation 33
4.4.3 Spatial Encoding via Graph Attention 33
4.4.4 Temporal Encoding via LSTM 34
4.4.5 Dense Reconstruction 35
4.4.6 Anomaly Score Computation 35
4.4.7 Score Fusion and Thresholding 36
4.4.8 Training and Inference Pipeline 36
4.4.9 Architectural Advantages 37

5 Result Analysis **38**
5.1 Performance Evaluation 38
5.1.1 Evaluation Criteria (Metrics) 38
5.2 Result and Analysis 40
5.2.1 Confusion Matrix 40
5.2.2 F1 Score Comparison 41
5.2.3 ROC-AUC Comparison 42
5.2.4 Average Precision Comparison 43
5.2.5 PR Curve 44
5.2.6 ROC Curve 45
5.3 Analysis of the Design Solution 46
5.4 Final Design Adjustments 47
5.4.1 Limitations 47

6 Conclusion **48**
6.1 Summary of Findings 48
6.2 Contributions to the Field 48
6.3 Recommendations for Future Work 48

References **52**

List of Figures

1.1 Methodology Flowchart 5

4.1 Constructed Map for Traffic Simulation using SUMO 22
4.2 Proportion of normal and anomalous events (left) and breakdown of anomaly types (right), with slow-downs being the most frequent . . . 26
4.3 Autoencoder Model Architecture 28
4.4 Isolation Forest Model Workflow 29
4.5 LSTM Model Structure 31
4.6 Overview of the proposed GATFA architecture. 32

5.1 Confusion matrices of different anomaly detection models. Subplot (a) corresponds to ARIMA, (b) to AutoEncoder, (c) to DBSCAN, (d) to Isolation Forest, (e) to K Means, (f) to LSTM AutoEncoder, and (g) to the proposed GATFA model. Each subplot shows counts of True Negatives, False Positives, False Negatives and True Positives 40
5.2 Performance Comparison of Unsupervised Models using F1 Scores . . 41
5.3 Performance Comparison of Unsupervised Models using ROC-AUC value 42
5.4 Performance Comparison of Unsupervised Models using Average Precision value 43
5.5 Precision–Recall Curves of Unsupervised Models highlighting performance under class imbalance 44
5.6 ROC AUC Curves of different unsupervised models showing trade-offs between True Positive Rate and False Positive Rate 45

List of Tables

4.1 Feature descriptions of the simulated traffic dataset. 23
4.2 Comparison of baseline and proposed models for traffic anomaly detection. 27

Chapter 1

Introduction

1.1 Background

Intelligent transportation systems (ITS) have emerged as a critical focus in modern urban planning especially with the advent of the Internet of Things (IoT) and the evolution of artificial intelligence. These systems are designed to manage traffic flow and safety,reduce congestion and optimize road conditions by employing smart technologies that seamlessly integrate data from vehicles and road infrastructure. Vehicle road collaboration systems are on the lead in this case. It includes real-time data exchange between vehicles and their surrounding environments. But managing and processing the vast data generated in real time poses a significant challenge particularly in anomaly detection as it is crucial to prevent accidents and ensuring smooth traffic flow.

1.2 Rationale of the Study

Traditional approaches to anomaly detection rely heavily on the old grid-based uniform traffic network structures which often fail to accommodate the complexity and irregularity of various road networks. Such models treat traffic data in isolation which ignores the interdependent nature of the model across different road segments. In recent findings, graph-based representations of road networks have been explored to better capture these interactions. This lead to the adoption of machine learning techniques for processing traffic data. Our research aims to address these limitations by proposing an enhanced graph-based anomaly detection method for an IoT-enabled vehicle-road collaboration systems. By leveraging the power of graph neural networks and optimizing sampling techniques, we aim to provide an objectively better method for detecting anomalies in traffic. Our approach shifts from grid-based models to a more dynamic,flexible and comprehensive graph-based framework, accounting for all sorts of complexity of road networks. This allows us to capture even the slightest interdependencies between road segments and traffic conditions which leads to improved anomaly detection performance. The proposed system is built by integrating self-supervised learning which reduces the need for immense manual data labeling. By combining numerical data from sensors on vehicles and road infrastructures,the system can continuously monitor traffic and identify irregularities that may indicate potential hazards like accidents,blockages,speeding etc. Our method also introduces a contrastive learning approach which enables the

model to more accurately detect anything out of the norm by comparing various subgraphs of the road network. Additionally,we propose an intimacy score system that will optimize subgraph selection, ensuring that the most relevant data is used for anomaly detection. Ultimately, the research seeks to contribute to the growing field of AIoT in transportation by greatly improving the efficiency of real-time anomaly detection which will potentially pave the way for smarter and safer road systems. Enhancment in scalability,efficiency and accuracy of graph-based models can represent a significant advancement in intelligent transportation technologies. It also offers a practical solution to the increasing complexity of modern urban traffic networks which people face everyday.

1.3 Problem Statement

The issue of efficiently monitoring and controlling traffic irregularities in intelligent transportation systems is still a major research challenge as those systems continue to deliver IoT based solutions for real-time traffic management. The road networks of metropolitan cities are expanding at a very fast pace with rapid urbanization coupled with an increase of vehicles plaguing the road which makes the dynamics of traffic flow more complicated. This brings forth several challenges such as: the problem of locating irregularities in the path of vehicles due to the complexity of the network structure,robust detection of possible traffic disturbance and road safety without putting too much strain on the computational framework.

Traditional grid-based approaches fail to address these problems adequately due to their rigid structure which assumes that road networks and traffic flows are uniformly distributed. In reality, urban traffic systems are highly irregular, with intricate relationships between road segments that cannot be easily captured by grid models especially in south Asian region. Moreover, traffic anomalies are often localized events, requiring a system that can detect such irregularities with high granularity and precision.

The lack of efficient, scalable and accurate anomaly detection methods presents a major obstacle to the full realization of intelligent transportation systems. A more refined approach is needed. One that can process complex, graph-structured data in real-time while maintaining high detection accuracy. Furthermore, the reliance on computer vision frameworks such as OpenCV for anomaly detection is computationally expensive, demanding significant processing power and memory. In contrast, leveraging numerical sensor data is more lightweight, making it better suited for real-time deployment on resource-constrained hardware.

Another key challenge lies in capturing the temporal complexity of traffic flow. Vehicle movements and road interactions evolve dynamically over time where small disturbances such as sudden braking or lane changes can propagate into larger anomalies like traffic jams or accidents. Traditional models often treat traffic data as static snapshots, overlooking these sequential dependencies. Therefore, an effective anomaly detection framework must not only account for spatial road-network struc-

tures but also integrate temporal dynamics to model traffic behavior as it unfolds in real time.

In parallel, the vision of smart cities increasingly relies on the integration of Artificial Intelligence of Things (AIoT) where interconnected sensors,vehicles and infrastructure work collaboratively to deliver intelligent services. Embedding AI into IoT ecosystems enables traffic management systems to be predictive and adaptive, providing real-time responses to anomalies with minimal human intervention. Nonetheless, bringing this vision to life requires anomaly detection techniques that are both computationally effective and sufficiently scalable to manage the extensive data streams produced by urban AIoT infrastructures.

By leveraging the strengths of graph neural networks and optimizing, the proposed method seeks to enhance the scalability and precision of existing models. Additionally, the research explores self-supervised learning techniques to eliminate the need for large, labeled datasets which reduces the overhead associated with manual data labeling and allowing the system to learn directly from the available data. The overarching goal is to create a robust anomaly detection system capable of operating efficiently in real-world traffic environments, ensuring that intelligent transportation systems can meet the demands of growing urbanization. In doing so,we address the drawbacks of current approaches and propose a solution that is both practical and scalable, paving the way for more advanced and dependable traffic management systems.

This research aims to address the limitations of traditional anomaly detection approaches by introducing a graph-driven, temporal vehicle–road collaborative framework by capturing complex spatiotemporal dependencies while remaining computationally efficient thereby meeting the real-time demands of intelligent traffic systems.

1.4 Research Objective

This research aims to develop an efficient and scalable anomaly detection framework for intelligent transportation systems that leverage vehicle-road collaboration through the integration of Augmented Intelligence of Things (AIoT). This system will be designed to improve real-time traffic management and road safety by accurately identifying anomalies in vehicle trajectories.

The primary objectives of this research are:

- **Develop a graph-based representation for the modelling of traffic networks:**
  In contrast to conventional models that employ a grid system, this study proposes that the road net is to be regarded as a topology graph where node elements are road segments and edges are their connections. With this graph representation, the system is able to understand the inter dependencies between different regions of the street network as well as the movement across regions which makes it more optimal in abnormality detection in the given context in real time.

- **Enhance anomaly detection using graph neural networks (GNNs):**
  GNNs are particularly well-suited to processing graph-structured data, such as traffic networks. By leveraging the capabilities of GNNs, this research aims to improve the detection of anomalies in vehicle trajectories, particularly in complex urban environments where traffic flows are highly interdependent.

- **Introduce a multi-view contrastive learning mechanism:**
  This research proposes a contrastive learning approach that compares multiple views of the traffic network to detect anomalies. By generating multiple sub-graphs around target nodes and comparing them with the target node's behavior, the system will more accurately identify deviations from normal traffic patterns. This method is anticipated to surpass conventional machine learning techniques in terms of accuracy and efficiency.

- **Implement a confidence score for optimized batch selection:**
  One of the key parts of this research is the introduction of an confidence score that can measure the relevance of neighbour nodes to target nodes in a graph-based framework. This score will help in the selection of the most informative portions of the network in anomaly detection procedures. This approach is expected to enhance system efficiency while reducing computational expenses.

The focus of this research is to enhance the technique of detection of traffic anomalies by making it more accurate, scalable and modular. These objectives, once achieved will perform in developing intelligent transport systems and therefore make the road networks safer and more effective in the context of emerging cities.

1.5 Methodology in Brief

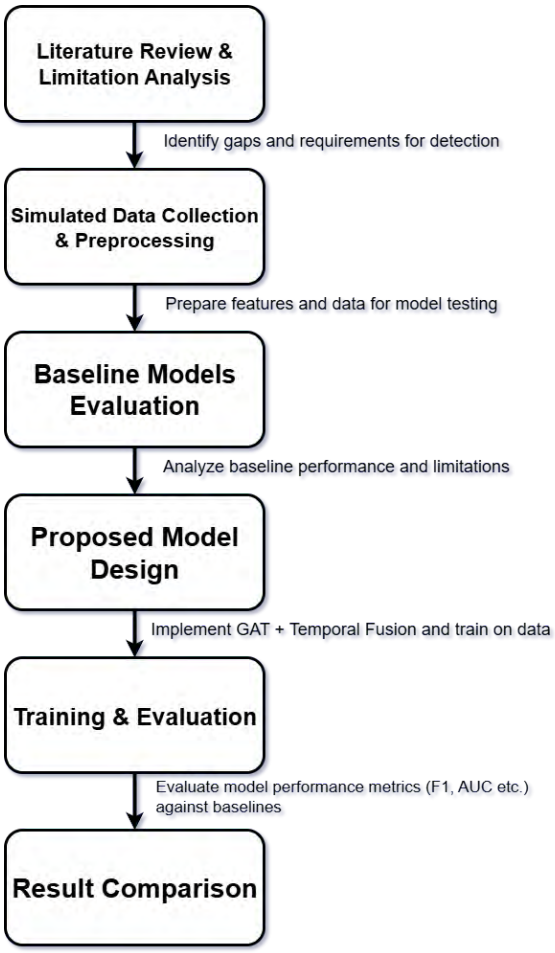

Figure 1.1: Methodology Flowchart

Fig. 1.1 illustrates the structured **six-phase methodology** employed for the development and rigorous evaluation of the proposed model. The process initiates with a comprehensive **Literature Review and Limitation Analysis** to establish critical research gaps and detection requirements. This is followed by **Simulated Data Collection and Preprocessing** to construct the foundational feature set using SUMO and Open StreetMap. A crucial **Baseline Models Evaluation** step establishes performance benchmarks against which the **Proposed Model Design** is validated. The final phases involve dedicated **Training and Evaluation** using standard performance metrics such as the F1 score and the Area Under the Curve (ROC AUC)which culminates in a definitive **Result Comparison** to quantify the proposed model's efficacy and superiority over established baselines.

1.6 Scopes and Challenges

1.6.1 Scopes of the research

This research focuses on the advancement in the techniques of the identification of anomaly in traffic systems by incorporating a graph-based framework combined with unsupervised and self-supervised learning techniques. This gives numerous scopes

- **Comprehensive Benchmarking:** Evaluation of conventional unsupervised models, for instance, Isolation Forest, LSTM, DBScan, KMeans and Arima, on a synthetic dataset generated by SUMO, to ensure a robust baseline for anomaly detection performance.
- **Graph-Based Modeling:** Representation of traffic networks as graphs, with road segments as nodes and connectivity as edges, ensure the spatial and temporal dependencies capture more effectively than conventional models.
- **Self-Supervised Learning:** Developing a model that learns to detect anomalies eliminating the dependency on labeled data by using contrastive learning to compare traffic patterns across the network that enables scalability to diverse traffic scenarios.
- **Adaptability:** To ensure the generalizability across different urban environments, the design of a flexible framework which is capable of adapting to various traffic network structures and anomaly types.

1.6.2 Challenges of the Research

Several challenges were identified while the methodology was being developed.

- **Synthetic Dataset Limitations:** While synthetic data allowed for controlled environment and large-scale experimentation, it lacks the full extensive variability and unpredictability of real-world traffic environments. This creates a potential gap which requires careful validation and domain adaptation strategies when transferring the approach to real deployments.
- **Class Imbalance:** The synthetic dataset demonstrates a low anomaly rate which creates a class imbalance.
- **Spatial Dependencies:** Capturing complex spatial relationships in dynamic road networks proved difficult for traditional models which motivates the shift to a graph-based approach but requiring sophisticated graph construction and analysis.
- **Temporal Dependencies:** Modeling the evolving patterns of traffic over time posed challenges for conventional methods that mandates the integration of temporal fusion techniques to effectively capture dynamic changes and ensure accurate anomaly detection.

Chapter 2

Literature Review

2.1 Preliminaries

As time advances, the globe is always introduced with new challenges and hurdles. In this century, the growth rate of population and the tendency of people moving to urban areas is one of the key challenges that the globe is facing. As a result, a gigantic amount of people reside in a small area causing various problems. Traffic anomaly is one of the byproducts of this issue. The necessity of accurate traffic anomaly detection has become indispensable for ensuring a smoother, faster and safer communication of urban areas. This literature review will be focusing on the findings from the latest research domains based on the traffic flow prediction, anomaly detection frameworks and intelligent transportation system and the synthesization of all the gathered information. Additionally, our research contributes to the field through the implementation and evaluation of seven unsupervised machine learning algorithms (DBSCAN, K-Means, LSTM, ARIMA, Isolation Forest and AutoEncoder) along with our proposed model on a synthetic dataset simulated by SUMO.

2.2 Review of Existing Research

Even though a large portion of the research in traffic analysis is focused on flow prediction, they are also in parallel with anomaly detection strategies in terms of our proposed methodology. In paper [1],[2], the ST-SSL framework which is Graph Neural Network(GNN) was used to overcome the limitations of previous traffic flow prediction models. This particular framework reached a new peak by the effective aggregation of data across time and space, this particular model combines both spatial and temporal convolutions. which, while forecasting traffic flow, guarantees its generalizability across many locations and time zones. This specific model also employed a self-supervising technique, which enables it to generate its own supervisory signals by predicting portions of the input data from other portions. It can learn from unlabeled data or more intricate traffic patterns with the aid of this technique. The framework outperformed other baseline methods in terms of both Mean Average Error (MAE) and Mean Average Percentage Error (MAPE) across all datasets. The effectiveness of GNN-based approaches wholly depends on their ability to model complex spatial relationships inherent in traffic networks while simultaneously capturing temporal dependencies. The ability of combining both spatial and temporal

convolution across multiple dimensions makes them suitable for traffic anomaly detection, as anomalies often create deviations in both spatial and temporal patterns.

In the paper [3], the Radial Basis Function(RBF) neural network was combined with Fuzzy logic which makes this model able to predict traffic speed during peak hour by incorporating nonlinear factors and external variables like weather and holidays. Which traditional linear prediction model fails to do.The Mean Absolute Error (MAE) was reported as 1.89 and the Root Mean Square Error (RMSE) as 2.15, which are considerably lower than the errors from other methods like the time series method (MAE: 11.39, RMSE: 13.25), BP neural network (MAE: 4.67, RMSE: -6.29) and RBF neural network (MAE: 3.96, RMSE: 5.36). Also it shows great progress in Mean Absolute Percentage Error (MAPE) in this sphere the hybrid algorithm was 6.4% while time series was 34.16% and BPNN and RBFNN was 14%.

The CoLA framework was used in [4], to detect several anomalies in networks with both structural and attribute information. It employed a Contrastive Self-Supervised Learning framework which works with local subgraph instances for anomaly detection. As a result, it can maintain efficiency while processing large networks,avoiding the memory and computational constraints of graph training. This model also outperformed all baseline methods achieving an average improvement of 6.44% in AUC scores.

As proposed in paper[5] driver will get the optimized road with less traffic flow information , the paper [6] used Multi-Agent Adaptive Routing Optimization(MAARO) which provides the driver optimized road by which he can reach a destination facing less traffic congestion. It is basically a decentralized system, where each node(e.g.,system,device) are independent of each other and make decisions on their own rather than a central authority controlling the network. Moreover, paper [5] focused on the sustainable transportation system that was achieved by paper [6] by reducing fuel consumption and greenhouse gas emissions with the help of the MAARO framework. The paper[7], proposed some algorithms to improve the traffic management system. For minimizing the congestion, intelligent traffic guidance systems using adaptive routing and big data analytics were proposed. Also, adaption of light phases based on real-time conditions and emergency vehicle prioritization by the dynamic traffic light control to reduce delay at intersections was proposed. In paper[8], the sensing technology was implemented. Different kinds of sensors were used in this model to sense any kind of anomaly like driver fatigue,impaired driving prevention etc to enhance the safety of transportation.

Paper[9] showed the necessity of computer vision to detect anomalies in public spaces,particularly on roads. The paper[10] implemented a model with the help of various kinds of library tools of OpenCV to make it capable of detecting and recognizing circular traffic signs in real-time using color and shape recognition. This model integrated multiple techniques like Combining color detection in the HSV space, Canny edge detection and Hough Transform for shape detection, for higher accuracy rate CNN. The use of CNN achieved impressive results(97-100% accuracy).

In the paper[11], it addressed the challenge of accurate reconstruction of vehicle trajectories in the urban intersection. It found out that, due to strong air flow and lower video resolution the UAV(data source) struggles to collect accurate data. With the help of wavelet transforms, which is an outlier detector it identified and removed outlier data from the trajectory data. And then interpolated the missing data by using Locally Weighted Regression (LWR) with Gaussian kernels. Then Savitzky-Golay filter was applied to smooth the velocity and direction angle data.

In the paper[12], it addressed the current challenges that Intelligent Transport Systems (ITS) and the Internet of Vehicles (IoV) faces like Inadequate Information Delivery, Limited Surveillance Infrastructure, Safety and Privacy Concerns etc. For tackling this issue they utilize the R-CNN model which is pioneering deep learning model for object detection having strength combining CNN and region proposal method. Another paper[5] stated that bringing AI into the transportation system will create a challenge in the field of privacy and safety. For that particular reason, paper[12] introduced federated learning which is a decentralized learning algorithm which allows every device to train their model on their own data and share the model update with the central data. That's how privacy of each device is maintained. Paper[13] addressed the major security concerns in the protocol that current electric vehicle (EV) charging infrastructure is using. One can easily eavesdrop, tamper data and inject command attacks into this system. They introduced Command-Level Anomaly(CAD) method to solve the problem. Firstly, they introduced dynamic analysis with fuzzy testing to analyze the information obtained from the protocol. Based on this analysis, A state transition diagram based on an MTM was designed to map out the deviated protocol messages. Then it applies Long Short-Term Memory (LSTM) which is a model well-suited for capturing complex, dynamic behaviors in anomaly analysis. LSTM has been effectively used in large scale real-time applications. The pipeline effectively detected anomalous behaviour, known vulnerabilities and discovered new vulnerabilities.

In paper [14], authors have come up with a modified K-Means clustering algorithm in order to overtake the limitations of the existing K-Means clustering model. Traditional K-Means algorithms heavily rely on the initialization and a priori specification of cluster numbers. The proposed algorithm achieves better performance due to automatic determination of optimal numbers of clusters without initialization of parameter selection with the help of an entropy based learning schema. It can be generalized across diverse datasets varying shapes and volumes which is ensured by robust clustering technique. The U-k-means algorithm employs a novel objective function with entropy penalty terms to facilitate competition among clusters, enabling it to discard redundant clusters dynamically. U-k-means clustering algorithm is better than conventional clustering algorithms like K-Means algorithm, X-Means, RL-FCM in terms of accuracy across all the tested datasets.
Clustering-based methods are often employed for anomaly detection, where anomalies are identified as data points that do not fit well into any cluster or lie far from dense cluster centers. In this context, an improved clustering approach such as U-k-means can enhance anomaly detection performance by producing more robust and well-separated clusters, thereby making deviations or outliers easier to identify. This

principle aligns with our research, where the objective is also to detect irregularities in complex traffic patterns, though through a graph-driven, temporal framework rather than purely unsupervised clustering.

Authors have proposed a hybrid machine learning algorithm consisting of K-means, Density-Based Clustering (DBSCAN) and Sequential Minimal Optimization (SMO) in the paper [15], to address the issue of network security in traditional anomaly detection methods. This model achieves superior performance by effectively incorporating clustering and classification techniques, which mobilizes K-means and DBSCAN for unsupervised clustering of network data and SMO for optimized classification. The robustness of anomaly detection is made sure by the model through rigorous identification of all the hidden patterns and network intrusions across diverse datasets. As mentioned earlier, the model employs a hybrid approach that enhances detection rates while focusing on the minimization of false positives, which ultimately improves the overall intrusion classification accuracy. The proposed method outperformed baseline approaches, achieving lower False Alarm Probability (FAP) and higher accuracy in terms of Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) across tested datasets.

The authors have come up with the idea of introducing hyperplanes with random slopes for data slicing in the existing Isolation forest model to eliminate its limitations of biases which occur due to axis-parallel branching, leading to artifacts in score maps in the paper [16]. Robust anomaly detection is ensured by accurately capturing data structures across diverse datasets, including synthetic (e.g., Gaussian blobs, sinusoidal data) and real-world benchmarks. The algorithm advances in two approaches: random data rotation before tree construction and, preferably, random-slope hyperplane branching; the artifact issue is solved here. EIF performs better than the standard IF, with a higher Area Under the Receiver Operating Characteristic (AUROC) and Area Under the Precision-Recall Curve (AUPRC) with lower score variance across tested datasets.

The standard deep autoencoders struggle handling noisy and outlier corrupted data. It can not make any progress without having clean training data. This issue is addressed by the authors in paper [17], they took inspiration from the Robust Principal Component Analysis by splitting input data (X) into (X = L_D + S), where (L_D) is reconstructed by a deep autoencoder and (S) captures noise and outliers. Robust anomaly detection is ensured by isolating anomalies into (S), enabling high-quality non-linear feature discovery across diverse datasets, such as noisy MNIST. The model implements an anomaly-regularizing penalty, optimized via backpropagation, proximal methods and the Alternating Direction Method of Multipliers (ADMM). A generalized Group Robust Deep Autoencoder (GRDA) distinguishes structured anomalies (e.g., corrupted features or instances). RDA and GRDA outperformed baselines, achieving up to 30% lower error rates than standard autoencoders and a 73% higher F1-score than Isolation Forest in anomaly detection on MNIST.

In paper [18], the authors proposed a novel traffic flow forecasting method, SDLSTM-

ARIMA, to address the limitations of conventional methods, such as poor stability, high data requirements and limited adaptability in intelligent transportation systems (ITS). Introduction of traffic data time singularity ratio for the sake of adjustment of the dropout probability, mitigation of overfitting and the improvement of robustness is ensured by the enhancement of Long Short-Term Memory (LSTM) neural network in the method. Combined with the Autoregressive Integrated Moving Average (ARIMA) model, SDLSTM-ARIMA mobilizes uneven time intervals (1-hour for LSTM, 15-minute for ARIMA) to gain accurate short- and medium-term predictions, particularly during peak traffic periods like 6 AM. Uses of various embedded systems like Raspberry Pi, computer vision and the Storm platform for real-time, traffic flow detection and prediction is seen in the study. Experimental results on a British Columbia traffic dataset demonstrate that SDLSTM-ARIMA performs better than traditional ARIMA as expected and AR-RBLTFa methods, achieving lower Mean Absolute Percentage Error (MAPE) and higher accuracy in both holiday and non-holidays. High accuracy is ensured by the incorporation of systems like Raspberry Pi, computer vision and the Storm platform for real-time, traffic flow detection and prediction, reliability and scalability, offering broad application prospects for ITS.

In paper [19],the authors identified a major research gap in traffic forecasting with Graph Neural Networks (GNNs). The existing research ignores the fact that it can combine with multi-source data such as sensors, weathers, social events and multimodal inputs (e.g., text, video, GPS), keeping only focus on the GNNs that are vastly used for capturing the spatial and temporal dependency. Absence of such fusion is responsible for models limitation in accuracy, robustness and scalability, especially under sparse or uncertain data conditions. The authors come up with the proposal of a systematic review of GNN-based fusion methods, which is basically focusing on hybrid models where GNNs are integrated with CNNs, RNNs, LSTMs in a motivation to reduce the research gap. Fusion strategies are categorized at multiple levels such as feature-level, graph-level, spatio-temporal and multi-modal while case studies on models such as STGGAN, TrafficStream and PVCGN show the benefits across forecasting tasks. However, these models were applied into widely used datasets like PeMS and NGSIM resulting in that fusion-based GNNs is successful in the reduction of errors in a significant manner (MAPE, MAE, RMSE) and improve accuracy, with STGGAN reaching 98.75% accuracy on PeMSD8 and ETGCN showing better performance on long-term horizons.

In the papers [20],[21], the authors address one of the key aspects and arguably the most critical task for the intelligent transportation systems, the challenge of accurate traffic speed prediction in urban road networks. The struggle of capturing the complex spatial-temporal dependencies of traditional models like ARIMA and SVM was very much visible, especially when any unpredictable situation arises like traffic jam or road accident where congestion happens without following any patterns. Current deep learning approaches, including CNNs and LSTMs, improve performance but their tendency of treating road networks as grid instead of the inherent graph-structured topology lacks in correct predictions. Which occurs inefficiencies while the modeling of spatial correlations and long-term temporal dynamics, eventually

leads to inaccurate predictions during peak traffic hour. To tackle these shortcomings, the authors come up with a proposal of a hybrid framework that combines the message-passing ability of Graph Neural Networks (GNNs) with the memory mechanisms of LSTM units named Graph Long Short-Term Memory (GLSTM) model. The failure of capturing the original graph-structured topology issue is resolved by GLSTM preserving the original road network topology by representing it as a directed graph, where nodes represent road segments with attributes (speed, length, width) and edges capture connectivity features such as traffic lights and directions. By incorporating message-passing over road-graph structure, the model can propagate traffic influence across nodes, while the temporal component handles sequential dynamics. To process graph inputs and to be able to model the long-term temporal dependencies and spatial propagation patterns, a modified LSTM cell was developed. Implemented in an encoder–decoder framework, GLSTM maps variable-length historical graph sequences to hidden states and predicts future speeds across multiple connected segments. Experiments on a real-world dataset from Xi'an (Beilin District, September–November 2017) shows the success of GLSTM over baselines including MLP, Seq2Seq, NLSTM and GNN. Using taxi trajectory data aggregated into 5-minute intervals, GLSTM achieves significant improvements across evaluation metrics: MAE of 4.92 (5.2% better), MAPE of 0.35 (1.8% better) and RMSE of 6.81 (2.9% better). Case studies further show that GLSTM closely tracks ground-truth speed trends during peak hours, effectively leveraging graph structure for more accurate network-wide forecasting.

Though each of the papers achieved significant results than the previous models in their respective field, due to the dynamic nature of science, there is always a shortcoming. In paper [1] and [2], though it gained high accuracy in predictions, while implementing it in real time it may still face challenges concerning the processing time for predictions. Also this ST-SSL framework combined Graph neural network and Self-Supervised learning may need high computational resources and expertise. In the paper[3], the model training datasets were taken from a single place and single time frame. Which hinders its generalization process. Also it may include some external factors but significant factors like accidents, road construction and real-time traffic incidents are excluded. Liu et al. [4] proposed that the CoLA network is not fully generalized as the other networks like heterogeneous or spatial-temporal networks. Also, the use of GNN and Contrastive sampling makes it more complex than a simpler method. In paper[6], the model proposed by Ajayi et al. poses significant privacy concerns and risks of data breaches while handling real-time data. Also, the MAARO depends heavily on the infrastructure , which leads to high cost. Paper[7] also shares the same problems as Ajayi et al. in reserach[6]. Additionally, integrating various kinds of devices and systems may face compatibility issues. In the paper[8], the model is completely based on the sensors. Ultrasonic sensors that are used in the model can give false signals due to environmental noise. Also, MQ3 alcohol sensors may not reliably detect all scenarios of alcohol consumption. In the paper[10], it is mentioned this model can only detect circular traffic signs which restricts its acceptability to a broader range of sign shapes. Also, the system's performance is limited by the datasets it was trained on. In paper[11], while completing the reconstruction process it took 403 seconds to complete 2140 trajectories which is a very time consuming process. Also, the model's effectiveness is contingent on the

quality of the raw data. Any type of corrupted data can limit its robustness.In the paper[12], the object detection process is fully dependent on the video camera. If any kind of natural disaster occurs then the full R-CNN model will be affected and the object detection process will be hampered. Additionally, the system's detection process is maintained by camera and gps which has a very short range. According to Li et al.[13], the CAD method consumes a lot of time for generating the test case. Also, implementation of CAD method needs large amounts of computational resources and vast amounts of expertise. Also they didn't implement the idea on traffic systems rather it was just implemented on EV charging systems. In paper[19], the use of Fusion models is computationally expensive, the risk of overfitting with noisy or incomplete data still there and often lacks interpretability. Dependence on narrow datasets like PeMS also raises generalizability concerns, while scalability issues persist in large heterogeneous traffic networks. The use of RNNs in hybrid models introduces the vanishing gradient problem which limits their ability to capture very long-term dependencies despite improvements from LSTMs and GRUs. The study of [20], relies only on basic road attributes, excluding external factors such as weather, holidays, or accidents, which limits robustness under varying conditions. The dataset is restricted to a single district with only 28 segments, raising concerns about scalability and generalizability to larger networks. Computational scalability is also not fully addressed and GLSTM may suffer from issues such as over-smoothing in deeper layers.

When we analyse the research gaps in these paper, most of the models here have a common shortcoming. That is due to limitation of datasets or lack of diverse datasets which drives the model to a bias and restricts its generalizability. Algorithms proposed in paper [15], [14], [16], [17] can be computationally very intensive for which the performance in real time applicability in large scale traffic networks can be limited. The idea of Extended Isolation Forest in reserach[16], can be effective on synthetic and benchmark datasets (e.g., Gaussian blobs, sinusoidal data), but the model's validation on real-world traffic data is limited, which raises concerns about practical applicability. We have seen the uses of hardware in the paper [18] which may limit its scalability and adaptability to different hardware environments. Moreover, the issue of latency can be arised due to the processing of high-frequency real-time data. However, to solve the computational issue of paper[15], [14], [16],[17] we need to go for the optimization of the algorithms for instance implementing lightweight clustering variants and using processing frameworks in paper [15], deploying the algorithm of paper [14] , [17]in a distributed computing platform, using approximate hyperplane calculations or subsampling techniques to reduce computational overhead of the algorithm of the paper [16]. As proposed in paper[22] by Caetano et al., if we are having a shortage of diverse datasets we can focus on synthetic data. Synthetic data means fabricating artificial data which mimics real-world data but does not contain real world data. To avoid the problem of biases that can be arised in the algorithms of the paper [15], [14], [16], [17], [18] we can run the models on synthetic datasets generated by SUMO which will enhance its generalizability for being run on diverse datasets. The hardware dependency of paper [18] can be solved by developing standardized interfaces and protocols (e.g., MQTT) to make the embedded system compatible with various hardware platforms, consistent with your compatibility solutions for [7]. Additionally, we can implement parallel

processing and edge computing to reduce latency in the SDLSTM-ARIMA model, aligning with your parallel processing suggestions for [11]. Also, for improving data quality we can rely on IoT devices and traffic cameras. For solving the implementation problem in paper[1] and [2] we can develop software tools that simplify the STL-SSL framework which will provide a guided process for those who don't have deep technical knowledge. As the STL-SSL framework needs high computational resources we can optimize the algorithm and for hardware accelerations, we can use graphics cards. For solving the dataset problem of paper[3] we can rely on the two techniques mentioned above. And to incorporate the external factors we have to extend the model to include real-time data on traffic incidents. For solving the implementation problem of Liu et al.[4] we have to develop an user-friendly tool to shorten the complexity by which it will be more accessible to the user without deep technical knowledge. And to expand the generalizability of CoLA we have to conduct further research and run test CoLA in heterogeneous and spatial-temporal networks. For solving the security and privacy breaching in paper[6],[7] we have to develop a data governance framework which prioritizes user privacy and incorporates advanced encryption methods to secure data transactions. We have to propose government and private companies to upgrade the infrastructure to have a proper implementation of models proposed in paper[6] and [7]. In order to tackle the compatibility issues that were raised by Ouallane et al. in paper[7], we need to develop standard protocols and interfaces for communication between different systems. We have to use multiple types of sensors and implement regular calibration protocols to increase the reliability of sensors used in the model of paper[8]. Advanced alcohol detection technologies with a larger detection range and greater sensitivity should be used to reduce false positives and negatives. The form identification model from the work needs to be expanded to recognize geometrical signals beyond circles [10]. This can be done by training additional CNN models using datasets containing a wide variety of shapes. Zhao et al. [11] suggests exploring parallel processing or other effective techniques for reconstructing vehicle tracks to overcome the model's temporal limitation. Additionally, deep learning techniques or agent-based modeling can be considered to capture complex interactions at intersections. Advanced image processing techniques should be applied to the models presented in paper [12] to enhance object detection in challenging situations. Also we can expand our sensor range relying on Infrared or LIDAR Sensors to keep track of more things. The model that was used in paper[13], which will not only work on the anomalies on EV charging system but also help to find the anomalies in traffic system.

Graph-based learning methods have demonstrated significant promise in addressing the irregular and dynamic nature of traffic networks. One of the most influential contributions in this field is the Graph Attention Network (GAT) proposed in [23]. GATs introduce an attention mechanism that allows nodes to assign different importance weights to their neighbors during message passing. Instead of uniformly aggregating information from all adjacent nodes, GATs dynamically learn which connections are more informative. Zhang et al. [24] proposed a cooperative vehicle-road anomaly detection system that explicitly employs subgraph sampling. Their method constructs road topology graphs from vehicle trajectory data and applies a random walk strategy to sample candidate subgraphs for each target node. To refine this process, they introduced an intimacy score that evaluates the relevance of

each sampled subgraph, ensuring that only the most informative contexts are used in the anomaly detection task. Together, these works underscore the importance of selectively focusing on meaningful subgraph contexts in large-scale, real-world traffic networks. While GATs implicitly weight neighbors through learned attention, Zhang et al.'s [24] cooperative framework leverages explicit subgraph sampling with scoring mechanisms. Both approaches provide valuable insights for developing efficient, scalable and accurate graph-based anomaly detection models suitable for intelligent transportation systems.

2.3 Summary of Key Findings

The usage of ST-SSL in paper [1],[2], GLSTM in [20], [21] and GNN-fusion models [19] can capture complex traffic dependencies which outperform traditional ML and deep learning baselines in MAE, MAPE and RMSE. Hybrid approaches like RBF + Fuzzy logic in paper [3], SDLSTM-ARIMA in paper[18] and MAARO in paper [5] further incorporate weather, holidays and congestion patterns for better traffic speed forecasting and route optimization. However, as observed in [19], traditional deep learning approaches such as CNNs and LSTMs treat road networks as grid-structured topologies, while real-world networks are irregular and graph-structured. Moving towards graph-based modeling offers more accurate predictions, though many existing methods remain limited by dataset dependence, high computational cost and hardware constraints. To address some of these issues, a cooperative vehicle–road anomaly detection framework[24] was proposed that employs subgraph sampling with intimacy scores to select the most informative neighborhood contexts, improving scalability and precision. Similarly, Velicković et al. [23] introduced Graph Attention Networks (GATs), which implicitly achieve selective neighborhood aggregation by assigning dynamic attention weights to nodes, allowing models to focus on the most relevant road connections. These works collectively highlight that subgraph sampling and attention mechanisms are key strategies for enhancing both accuracy and efficiency in real-time traffic anomaly detection and forecasting.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

This research focuses on developing a real-time traffic anomaly detection system that leverages unsupervised learning within a vehicle-road collaborative environment. The system is designed to handle high-dimensional traffic data streams and is structured to support future AIoT deployments. The following summarizes the implemented components and design requirements:

- **Data Generation and Features:** Traffic data is generated using a SUMO simulation based on an OpenStreetMap (OSM) network. Roadside sensors are placed at each lane to capture contextual features including timestamp,vehicle speed,acceleration,lane occupancy,heading etc. over a 24-hour period, simulating a vehicle-road collaborative ecosystem.
- **Anomaly Injection:** To emulate real-world traffic disruptions, a controlled portion of sensor readings is programmatically injected with anomalies distributed across time and space. While unsupervised models do not require labels for training, an `anomaly_label` flag is maintained solely for evaluation purposes.
- **Base Models:** The framework integrates unsupervised learning approaches such as Autoencoders, LSTM Autoencoders, Isolation Forest, DBSCAN, KMeans and ARIMA. Models are assessed based on reconstruction error, clustering performance and predictive consistency, with F1-score and AUC serving as primary evaluation metrics.
- **System Architecture:** Implemented in Python, the pipeline employs Pytorch, Keras and Scikit-learn and follows a modular design that includes data preprocessing, normalization, anomaly injection, training and validation. The architecture supports streaming data to simulate real-time anomaly detection.
- **Graph-Based Modeling:** To incorporate spatial relationships and road network topology, traffic data is structured into graphs using NetworkX. This extension enables spatiotemporally-aware anomaly detection through Graph

Attention Network (GAT), Temporal Encoding (LSTM) and subgraph sampling which efficiently captures interactions between vehicles and road segments.

- **Outputs and Interfaces:** The system provides actionable outputs including anomaly score,label and lane report which supports both real-time monitoring and post-hoc traffic analysis.
- **Scope and Constraints:** The simulation spans morning, day and night periods to ensure temporal diversity. Dataset size and anomaly ratios are fixed to allow consistent benchmarking. The system is designed to be reproducible, scalable and operable on mid-tier hardware while remaining extensible for deployment in real-world AIoT environments.

3.2 Societal Impact

With cities expanding rapidly, traffic systems must evolve too. Our thesis supports safer roads and smarter urban planning.

- **Road Safety:** By identifying anomalies early, the system helps prevent accidents before they happen. These are not just warnings but potentially life saving interventions.
- **Emergency Efficiency:** Locating traffic disruptions accurately helps emergency services respond faster and reduces critical delays.
- **Smart City Integration:** Collected data helps planners and engineers in fixing bottlenecks, rework intersections and fine tune traffic lights using real behavior rather than assumptions.
- **Automation Benefit:** Unlike manual monitoring, this system if integrated efficiently can runs 24/7 which offers a more reliable and scalable traffic management approach.

3.3 Environmental Impact

Although not directly centered on sustainability, the system aids cleaner, greener cities in important respects.

- **Emission Reduction:** Smoother traffic means less idling and braking, which reduces fuel use and emissions over time.
- **Congestion Management:** Through early identification of traffic congestion, it diverts motorists prior to gridlock which can help alleviate city pollution.
- **Efficient Infrastructure:** The automated system minimizes the need for energy-heavy manual stations and reduces frequent maintenance.
- **Scalable and Sustainable:** As cities expand,the system grows through code and data without much increase in energy demands.

3.4 Ethical Issues

The development and deployment of an AIoT-based traffic anomaly detection system raise several ethical considerations and professional responsibilities that must be addressed:

- **Privacy and Data Security:** Since the system involves collecting and processing vehicle and infrastructural data, it raises concerns about user privacy. Ensuring that personal information, such as vehicle identities and driver behaviors, are anonymized and stored securely is crucial.
- **Bias and Fairness:** Machine learning models may inadvertently encode biases present in the data that can potentially lead to unfair treatment of certain regions or populations. Responsible data collection and model validation are necessary to minimize bias.
- **Safety and Reliability:** The system's decisions like identifying anomalies or faults directly affect traffic management and public safety. Ethical responsibility entails ensuring the system's accuracy, robustness and fail-safes to prevent harm due to erroneous detections.
- **Transparency and Accountability:** Developers should maintain transparency regarding how the system operates, especially in cases of false positives or negatives that might have safety or legal implications. Clear accountability structures must be established.
- **Professional Responsibilities:** Engineers and researchers have a duty to adhere to standards of honesty, integrity and societal welfare. Proper documentation, peer review and adherence to ethical research guidelines are essential.

3.5 Project Management Plan

Our process followed a structured, phase-based format to ensure that each component of the system was properly planned, constructed and tested. We began with a literature review and initial design choices, followed by dataset construction using SUMO and preliminary model testing. Tasks were assigned using a weekly sprint process with regular team meetings.The phases included model training, performance evaluation and visualization. Version control was managed using Git and progress was tracked using both online shared board and in-person meetings. This ensured timely delivery while allowing flexibility for adjustments.

3.6 Risk Management

We worked proactively to manage both technical and organizational risks.

- **Data Challenges:** Due to the unavailability of real-world anomaly data, we utilized synthetic inputs and self-supervised learning techniques.
- **Performance and Scalability:** To address the resource intensive nature of the models, we optimized the pipeline, modularized the codebase and trained models in mini-batches.
- **Deployment Readiness:** Docker was used to ensure compatibility and smooth execution across different environments.
- **Team Coordination:** Clear task division, consistent meetings and shared collaboration tools helped maintain project momentum and prevent delays.

3.7 Economic Analysis

The proposed traffic anomaly detection framework offers significant economic benefits through its computational efficiency and scalability. Its dual spatial-temporal graph-based model minimizes processing demands that enables deployment on low-cost hardware. This results in low infrastructure and operational costs. The system's capabilities can enforce traditional image-based methods by delivering faster detections with lower resource consumption in a multi-view architecture which reduces hardware and energy expenses and supporting sustainable urban traffic management.

A basic cost-benefit analysis indicates potential savings from reduced traffic congestion, fewer accidents and quicker emergency responses which translates to lower fuel consumption, shorter travel times and decreased vehicle emissions. Over time, these improvements can lead to substantial economic gains, with faster incident detection shortening congestion periods and boosting productivity. The system's efficiency suggests a favorable ROI with a short payback period and long-term savings which makes it an economical solution for smart city infrastructures. Moreover, the proposed model can provide valuable insights for city planning by identifying recurring traffic bottlenecks, unsafe road segments and regions with frequent anomalies. Such information can guide infrastructure investment decisions, helping authorities prioritize road expansions, redesign intersections or deploy adaptive traffic signals where they are most needed. By integrating these predictive insights into long-term urban planning, municipalities can reduce unnecessary construction costs, improve land use efficiency and design more resilient transportation networks. Consequently, the model not only supports real-time traffic management but also contributes to cost-effective, data-driven urban development strategies.

Chapter 4

Proposed Methodology

4.1 Phases of the Methodology

This research focuses on building an advanced framework capable of identifying anomalies in traffic networks while ensuring generalizability. The study uses a synthetic dataset generated by the SUMO simulator which consists of 70,777 records and 29 columns with a low anomaly rate. The proposed approach advances through five key phases, evolving from traditional unsupervised models to a novel graph-based, self-supervised framework.

4.1.1 Phase 1: Evaluating Traditional Models

The research begins with the implementation of conventional unsupervised machine learning algorithms on the synthetic traffic dataset. This dataset features realistic traffic scenarios including variations in flow,congestion and anomalies ultimately providing a controlled environment for model assessment. Six diverse mathematical models are employed for base evaluations. Each algorithm was chosen for its distinct approach to anomaly detection which allows a comprehensive analysis of strengths and weaknesses in traffic data. Preprocessing techniques address missing values,scaling, feature selection etc.

4.1.2 Phase 2: Recognizing Research Limitations

Although traditional models provide valuable insights, several limitations are observed in dynamic traffic networks. K-means and DBSCAN require extensive parameter tuning. ARIMA struggles with non-linear temporal patterns. Autoencoders may produce less discriminative reconstruction errors in high-dimensional data. LSTM is computationally intensive. All models struggle to capture spatial dependencies and are sensitive to low anomaly rates.

4.1.3 Phase 3: Adoption of a Graph-Based Framework

To overcome these limitations, the traffic network is represented as a graph $G = (V, E)$, where road segments are nodes $v \in V$ and connections are edges $e \in E$. This structure allows the model to capture traffic flow naturally and effectively by modeling both spatial and temporal dependencies.

Spatial Modeling: The graph captures spatial relationships between road segments which can help in tracking anomalies across neighboring segments.
Temporal Integration: Time-varying features are incorporated into nodes and edges which lets the model capture temporal dynamics alongside spatial dependencies.

4.1.4 Phase 4: Designing Spatio-Temporal Graph Based Model for Anomaly Detection

We propose a graph-based temporal autoencoder as the primary system for detecting anomalies in traffic data. This is a reconstruction based method that learns the typical patterns of spatial and temporal traffic dynamics. By training the model to faithfully recreate normal input sequences, it naturally generates higher reconstruction errors for unusual events, which we use as the anomaly score. **Encoding**

Spatio-Temporal Dynamics:
The model processes traffic features across a graph structure using a sliding window of length L over time. For each time step t, the input features X_t (representing N lanes with F features each) are handled in sequence.

Spatial Encoding: To capture immediate dependencies between lanes, we first apply a linear transformation to the raw features X_t, then pass them through a graph operation, specifically a mean aggregator in our implementation. This uses the graph's edge connections to produce spatially informed embeddings H_t^S for each time step, resulting in a sequence of these embeddings over the window.

Temporal Encoding: Next, we feed this sequence of spatial embeddings into a recurrent layer, such as an LSTM encoder, to model longer-term temporal patterns and condense the evolving traffic behavior into a compact context vector H_{temp}.

Reconstruction and Anomaly Detection: From the temporal context vector (after applying dropout for regularization), we use a simple decoder to reconstruct the original input sequence as $\hat{X}_{1:L}$. During training, we minimize the mean squared error between the inputs and reconstructions to embed the characteristics of normal traffic. During inference, we identify anomalies when the reconstruction error surpasses a dynamically modified threshold, indicating a deviation from accepted standards.

4.2 Dataset Description

The dataset used in this research was synthetically generated using the SUMO (Simulation of Urban Mobility) traffic simulator combined with OpenStreetMap (OSM) map data. The goal was to simulate a realistic urban traffic environment over a 24-hour period and inject both natural and anomalous behaviors for the purpose of anomaly detection research.

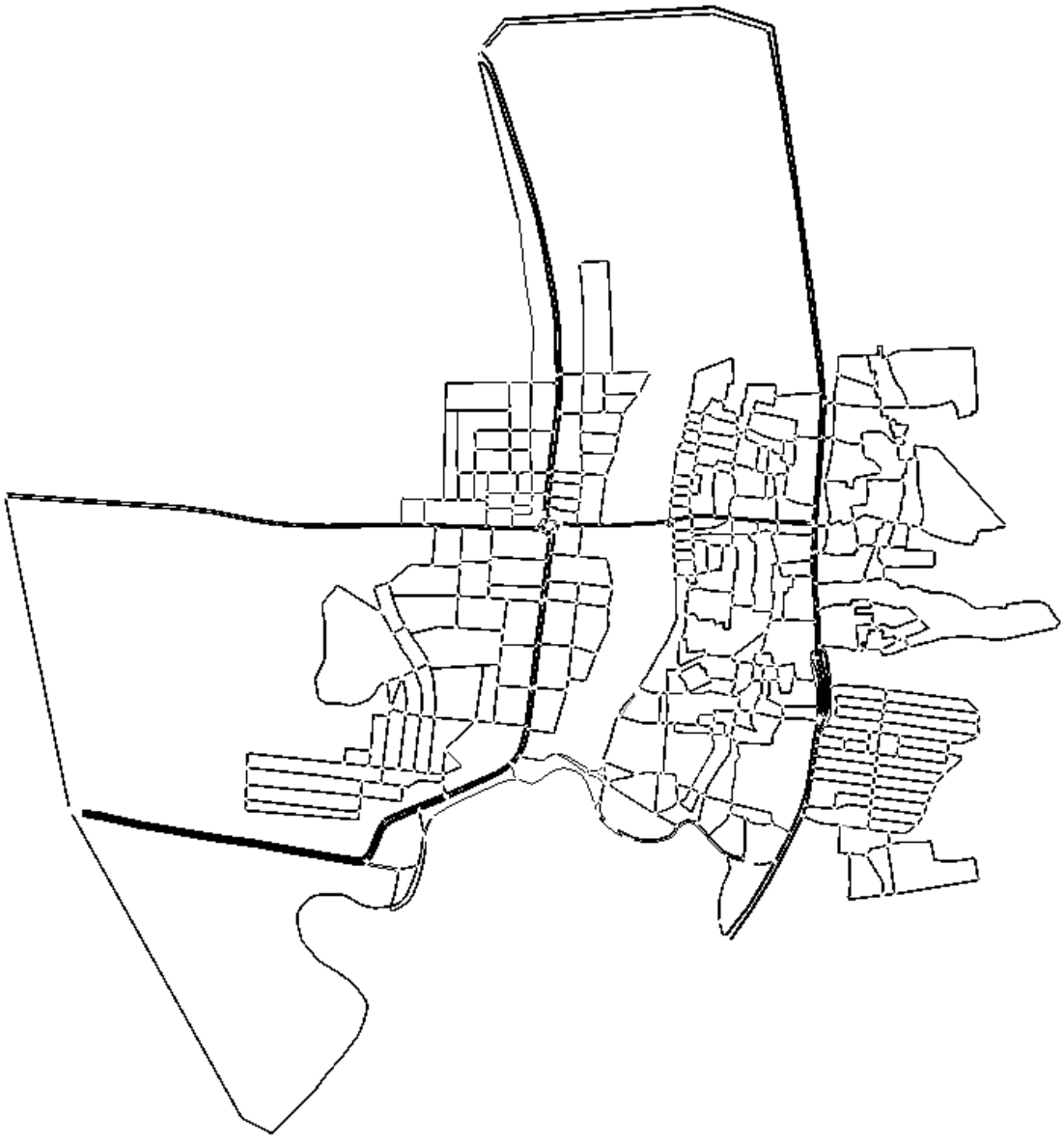

Figure 4.1: Constructed Map for Traffic Simulation using SUMO

4.2.1 Simulation Setup

- **Location:** A structurally complex network inspired from South Asian Road Structure. Base Map was extracted using OpenStreetMap.
- **Network Complexity:** The selected area features a dense and complex road network with various vehicle types, intersections, one-way streets and varying

lane configurations.

- **Sensor Placement:** Lane Area Detectors were placed on each lane segment to mimic real-world traffic monitoring systems. These act as virtual sensors that records traffic statistics at regular intervals.
- **Simulation Duration:** The simulation was run continuously for 24 hours by executing custom-modified trips according to the time of the day. Rush hours had more vehicle density than normal hours.

4.2.2 Feature Columns

The final dataset includes a total of 29 attributes, recorded at regular intervals from each lane-based sensor:

Category	**Features and Description**
Meta Information	*timestamp* (simulation time in seconds), *lane_id* (unique lane identifier).
Traffic Volume	*vehicle_count*: number of vehicles detected in the lane.
Speed Statistics	*mean/min/max/std_speedkmph*, plus *p10, p25, p50, p75, p90* percentiles describing speed distribution (km/h).
Acceleration Statistics	*mean/min/max/std_acceleration* of vehicles (m/s^2).
Flow / Density	*occupancy* (lane occupied time %), *flow_veh_per_hr* (extrapolated flow rate), *density_veh_per_km* (vehicles per km).
Delay / Waiting	*waiting_vehicles* (stopped vehicles), *mean/max_wait_time* (vehicle delay in seconds).
Other State	*halted_ratio*: fraction of vehicles with near-zero speed.
Lane / Network Context	*lane_index* (position within edge), *num_lanes_on_edge*, *edge_indegree/outdegree* (graph connectivity).
Anomaly Features	*anomaly_label* (0 = normal, 1 = anomaly), *anomaly_type* (e.g., accident, roadblock, slowdown, speeding, jam, stop, wrong lane).

Table 4.1: Feature descriptions of the simulated traffic dataset.

4.2.3 Anomaly Injection

The anomaly injection module was designed to introduce controlled disturbances into the simulation, as **SUMO** inherently models only smooth and normal traffic behavior. To enable the evaluation of anomaly detection models under realistic conditions, artificial anomalies were systematically incorporated to replicate irregular events such as vehicle breakdowns, lane blockages and sudden speed fluctuations. Depending on the anomaly type, specific effects were applied to vehicles or lane segments such as forced deceleration, random halts or partial lane obstructions, causing disruptions that propagate through the surrounding traffic. These injected anomalies create diverse and temporally dynamic irregularities, ensuring the simulation captures both normal and abnormal traffic patterns essential for robust model assessment.

- **Accident / Minor collision:** A subset of vehicles in the affected lane is forced to decelerate and stop which simulates a collision. The lane's maximum speed is reduced substantially that creates a localized bottleneck that can propagate congestion downstream.
- **Roadblock:** The lane is effectively blocked by setting its maximum speed close to zero. Vehicles in the lane are forced to halt or, if alternative paths exist, to reroute.
- **Slow down:** Vehicles on the affected lane have their speeds reduced. Both individual vehicle speeds and the lane speed limit are scaled down in proportion to the event severity, simulating temporary disturbances such as sudden braking or poor road conditions.
- **Speeding:** Selected vehicles are assigned speeds above the normal lane limit (up to a configured multiplier). This represents aggressive driving behavior and introduces local inconsistency in the flow profile.
- **Vehicle stop:** Randomly selected vehicles are brought to a complete halt for a short duration (e.g., a breakdown). Halted vehicles block following traffic and induce micro-congestion effects.
- **Wrong lane:** A randomly chosen vehicle is moved into an adjacent or incorrect lane (via lane change or repositioning). This creates lane-change irregularities and may trigger downstream disturbances including potential cascading accidents.
- **Traffic jam:** Vehicle speeds in the lane are reduced to approximately $0.3\times$ their normal values for an extended duration, representing sustained congestion.

The **Propagated Anomaly Injection** algorithm introduces a novel mechanism to simulate how local disruptions spread through the traffic network. Unlike static injections, it captures cascading effects across lanes and segments which produces more realistic anomaly dynamics for intelligent traffic systems.

Algorithm 1 Propagated Anomaly Injection in Traffic Simulation

Require: Simulation step t, edge-to-lanes mapping $edge\_to\_lanes$, propagation parameters

Ensure: Scheduled primary and secondary anomaly events

Select random edge e weighted by the number of lanes
Select random lane l from edge e
Select random anomaly type t from {accident, wrong_lane, roadblock, slow_down, speeding, vehicle_stop, minor_collision}
Sample severity $sev \in [0.3, 1.0]$
Sample duration dur
Create primary event $E_p = (id, t, t + dur, e, l, t, sev)$
Schedule(E_p)
{Propagate to other lanes of same edge}
for all lane $l' \in edge\_to\_lanes[e], l' \neq l$ **do**
    Create slowdown event E_s with reduced severity
    Schedule(E_s)
end for
{Downstream propagation via BFS}
Initialize queue $Q \leftarrow [(e, 0)]$, visited $\leftarrow \{e\}$
while $Q \neq \emptyset$ **do**
    Pop (e_{cur}, hop) from Q
    if $hop \geq 1$ **then**
        Compute delay $\leftarrow$ PROPAGATION_DELAY_PER_HOP $\cdot hop +$ random offset
        Decay severity $\leftarrow \max(0.2, sev \cdot 0.8^{hop})$
        Select type $\in$ {traffic_jam, slow_down, lane_block}
        Sample duration scaled by random factor
        Create propagated event E_{prop}
        Schedule(E_{prop})
    end if
    if $hop < MAX\_PROPAGATION\_HOPS$ **then**
        for all outgoing edge $e_{next} \notin visited$ **do**
            Add $(e_{next}, hop + 1)$ to Q and visited
        end for
    end if
end while
{Special case: wrong lane –¿ downstream accident}
if t = wrong_lane and random() < WRONG_LANE_TO_ACCIDENT_PROB **then**
    Select candidate downstream edge e'' within 1-3 hops
    Compute delay based on hop
    Select lane $l'' \in edge\_to\_lanes[e'']$
    Create accident event E_a
    Schedule(E_a)
end if
return All scheduled events

4.2.4 Statistical Summary of the Dataset

- **Total data points:** 70,777
- **Normal samples:** 69,305
- **Anomalous samples:** 1,472

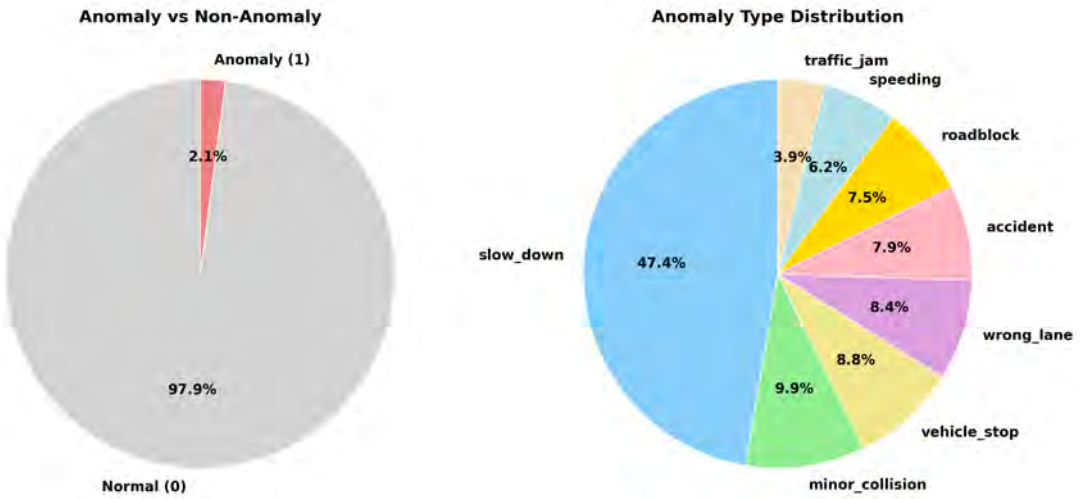

Figure 4.2: Proportion of normal and anomalous events (left) and breakdown of anomaly types (right), with slow-downs being the most frequent

This chart (4.2) shows the relatively low proportion of anomalous samples in the dataset is intentional and reflects realistic traffic conditions, where anomalies such as accidents, sudden slowdowns or lane violations occur infrequently compared to normal traffic flow. To further enhance realism, anomalies are not only injected at a single point but are propagated across time and space. For example, an accident on a lane can induce slowdown events in neighboring lanes or downstream edges. This mimicks the natural spread of congestion in a road network. The higher number of slow-down anomalies is also intentional, as minor decelerations frequently occur in response to traffic variations, temporary obstacles or the ripple effect of other anomalies, making them the most common type of detected anomaly.
Additionally, random jitter was introduced to numerical readings (such as vehicle speed or lane occupancy) to simulate natural variations in sensor measurements. This ensures that the synthetic dataset more closely resembles real-world sensor data, avoiding overly uniform or artificial patterns while preserving the statistical and spatial characteristics of anomalies.

4.3 In-Built Models

Model	Approach	Temp.	Spat.	Strength	Limitation
Autoencoder	Neural network reconstruction	✗	✗	Learns complex normal patterns	May misclassify rare-but-normal cases
Isolation Forest	Tree-based anomaly scoring	✗	✗	Handles high-dimensional data well	Sensitive to contamination rate
DBSCAN	Density-based clustering	✗	✗	Finds outliers without predefining clusters	Struggles with varying densities
K-Means	Centroid-based clustering	✗	✗	Fast and interpretable	Requires preset cluster number
LSTM Autoencoder	Sequence modeling with reconstruction	✓	✗	Captures temporal dependencies	Needs sequential data; costly training
ARIMA	Statistical time-series forecasting	✓	✗	Models trends and correlations	Assumes linear and stationary data
Proposed GAT + Temporal Fusion	Graph attention with temporal fusion	✓	✓	Captures spatial–temporal relations; ideal for lane-level anomaly detection	Computationally heavy; graph tuning required

Table 4.2: Comparison of baseline and proposed models for traffic anomaly detection.

To explore various strategies for traffic anomaly detection, we initially evaluated six key machine learning models on the synthetic traffic dataset generated using SUMO. As summarized in Table 4.2, these models were selected because each represents a distinct methodological approach to identifying anomalies. This comparative analysis provided valuable insights into how different algorithms interpret and flag unusual traffic behaviors or irregular patterns, forming the foundation for developing our more advanced graph-based spatio-temporal framework.

Autoencoder

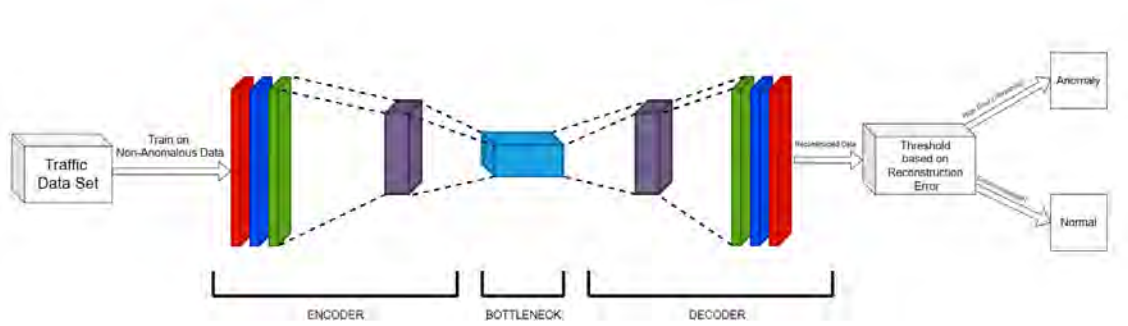

Figure 4.3: Autoencoder Model Architecture

Figure (4.3) illustrates the architecture of a base Autoencoder model, which learns to compress and reconstruct input data, enabling effective anomaly detection by measuring reconstruction errors. It is an unsupervised neural network architecture that is commonly used for image processing. It can also be used to learn compressed representations of normal traffic behavior. It consists of an encoder and decoder function for dataset reconstruction. In order to minimize the reconstruction loss, the model is trained. The reconstruction error has the following mathematical definition:

$$\mathcal{L}(x, \hat{x}) = |x - \hat{x}|^2, \tag{4.1}$$

where x is the input and $\hat{x} = g(f(x))$ is the reconstruction. After training on normal traffic samples,any input yielding a high reconstruction error (above a defined threshold) is considered anomalous. We used high dimensional latent space and evaluated the model using evaluation metrics such as F1-score, precision, recall, AUC and reconstruction error plots.

KMeans Clustering

KMeans clustering divides the dataset into k clusters by minimizing the total within-cluster variance. Each sample is assigned to the nearest centroid, optimizing the following objective:

$$\min_{\{\mu_j\}} \sum_{k=1}^{n} \|x_k - \mu_{c_k}\|^2, \tag{4.2}$$

where x_k represents sensor data record in the traffic dataset, μ_{c_k} is the centroid of its assigned cluster and n is the total number of samples. In this study, the efficient number of clusters was found to be 10 using the Silhouette Coefficient.Then A hybrid anomaly detection mechanism that combining cluster-level anomaly ratios and distance-based thresholds was applied to enhance detection robustness.

Isolation Forest

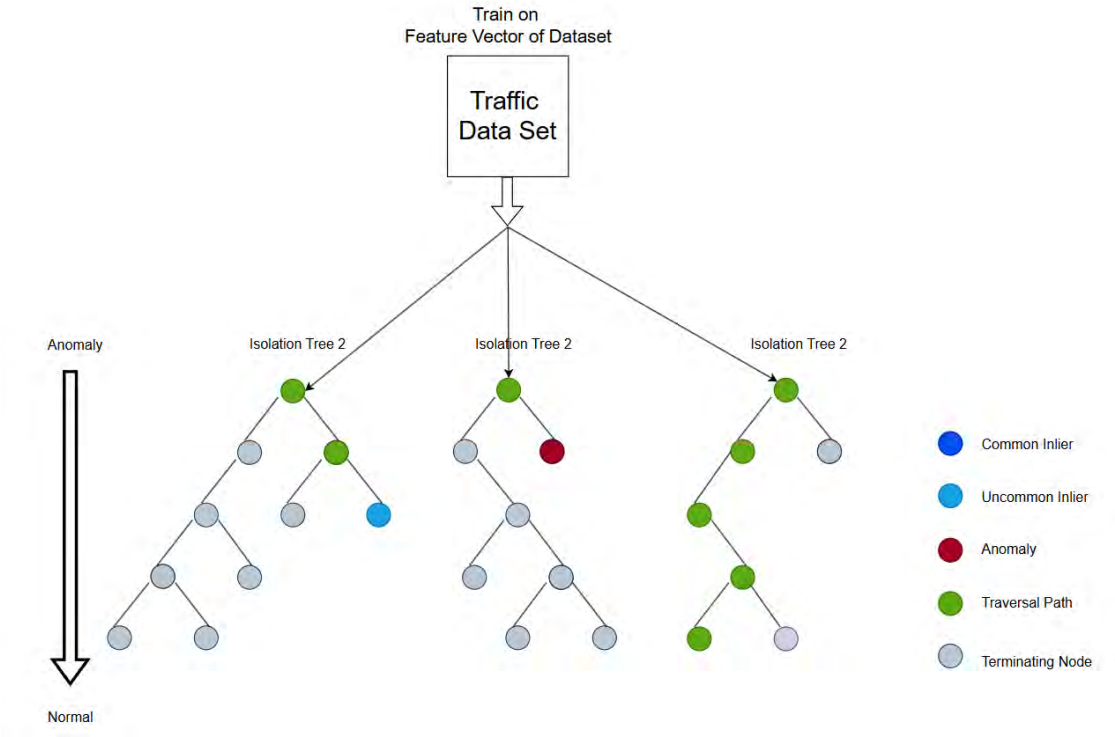

Figure 4.4: Isolation Forest Model Workflow

Figure (4.4) presents the Isolation Forest workflow, where anomalies are detected based on how quickly instances are isolated in randomly generated trees.
Isolation Forest identifies anomalous samples by recursively partitioning the feature space using randomly selected attributes and split values. Anomalies are usually classified with fewer splits than normal due to their unique properties.
The anomaly score for a data point x is defined as

$$s(x) = 2^{-\frac{E(h(x))}{c(n)}}, \tag{4.3}$$

where $E(h(x))$ denotes the expected path length required to isolate x and $c(n)$ represents the average path length of unsuccessful searches in a binary search tree of size n, approximated as

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}, \tag{4.4}$$

with $H(i) = \ln(i) + \gamma$ denoting the i-th harmonic number and $\gamma \approx 0.5772$ being the Euler-Mascheroni constant. A score $s(x)$ approaching 1 indicates a higher likelihood of anomaly. In this study, features such as speed, acceleration, vehicle count and occupancy were normalized prior to model training. Model performance was assessed using confusion matrices and standard classification metrics.

DBSCAN (Density-Based Spatial of Application Clustering with Noise)

DBSCAN recognizes clusters on the basis of dense regions of points distinguished by the area of lower density.DBSCAN doesn't hold the necessity of declaring the number of clusters beforehand. It is in need of two parameters: the radius ε and the minimum number of points $MinPts$. A point will be identified only as non anomalous when it gets placed within a cluster's ε neighbourhood which achieves the quota of minimum number of points $MinPts$.:

$$N_\varepsilon(p) = \{q \in D \mid \|p - q\| \leq \varepsilon\}. \tag{4.5}$$

In this experiment, the optimal parameters were selected using the k-distance method and F1-based tuning which allows the mathematical model to capture dense traffic clusters while labeling sparse, irregular data points as anomalies.

ARIMA (AutoRegressive Integrated Moving Average)

The AutoRegressive Integrated Moving Average (ARIMA) model is used with the motivation of capturing the temporal relationships present in the traffic dataset. This model is a integration of autoregressive (AR) elements where differencing ensure stationarity and moving average (MA) components that eventually enable it to effectively capture linear trends and dependencies in time-series data. In general, the ARIMA model can be mathematically formulated as:

$$\Phi_p(B)(1 - B)^d X_t = \Theta_q(B)\epsilon_t, \tag{4.6}$$

where B is the backshift operator, $\Phi_p(B)$ and $\Theta_q(B)$ represent the autoregressive and moving average polynomials of orders p and q, d denotes the degree of differencing and ϵ_t is the white noise error term. In this experiment, an ARIMA(2,1,2) configuration was fitted individually to the top-ranked features selected using the ANOVA F-test (`SelectKBest`). The model forecasts were compared against actual observations to compute residuals, which quantify the deviation between predicted and observed behavior. Samples with residual magnitudes exceeding a dynamic threshold, defined as $0.9 \times \sigma_{train}$ (where σ_{train} is the residual standard deviation from the training set), were labeled as potential anomalies.
A weighted aggregation scheme was then applied across all monitored features to compute the final anomaly score, with a threshold of 0.10 determining anomalous segments. This hybrid temporal-statistical approach enabled the detection of irregular traffic dynamics that deviate from expected historical patterns while maintaining high interpretability and stability in results.

LSTM Autoencoder

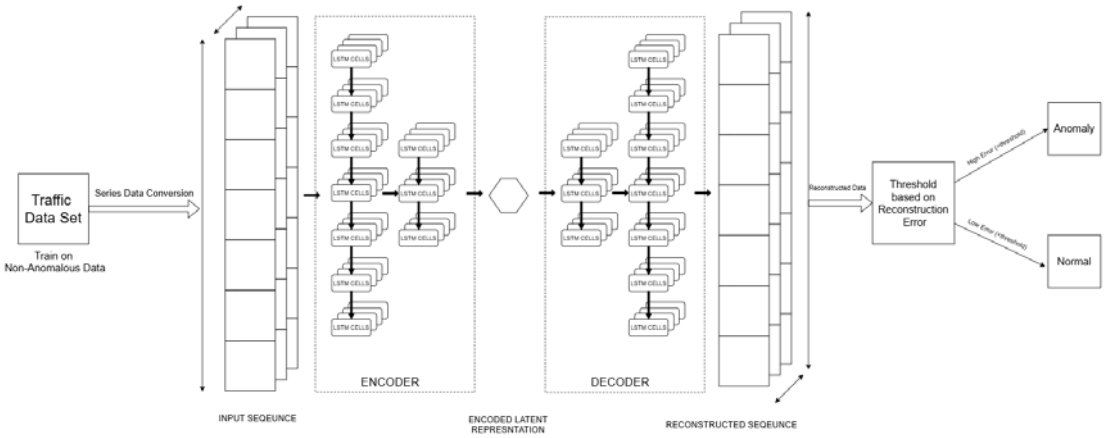

Figure 4.5: LSTM Model Structure

Figure (4.5) depicts the LSTM model structure used to capture temporal dependencies in sequential data which makes it suitable for detecting anomalies over time. The LSTM Autoencoder captures temporal dependencies in sequential traffic data. Input sequences of 30 time steps were passed through an LSTM encoder to obtain hidden states h_t and then reconstructed by a decoder. The reconstruction error over a sequence of length T is defined as

$$\mathcal{L}_{\text{seq}} = \frac{1}{T} \sum_{t=1}^{T} \|x_t - \hat{x}_t\|^2, \tag{4.7}$$

where x_t is the input at time step t and $\hat{x}_t$ is the corresponding reconstruction.Which was used to detect anomalies by thresholding above the 99th percentile. This method is particularly effective for capturing evolving traffic anomalies like gradual congestion or abrupt disruptions.

4.4 Proposed Model Specification

The GATFA model leverages graph-based spatial encoding, sequence-based temporal prediction and dense reconstruction to generate anomaly signals, which are fused to identify deviations from normal traffic patterns. Unlike conventional spatio-temporal autoencoders that often rely on a single encoding-decoding pathway, GATFA employs independent modules for each aspect, allowing for specialized learning of spatial, temporal and reconstructive features. The model can capture intricate interactions in traffic data thanks to its modular approach, such as lane connectivity (spatial), historical trends (temporal) and overall data consistency (reconstruction), providing a robust framework for detecting both localized incidents (e.g., accidents) and widespread disruptions.

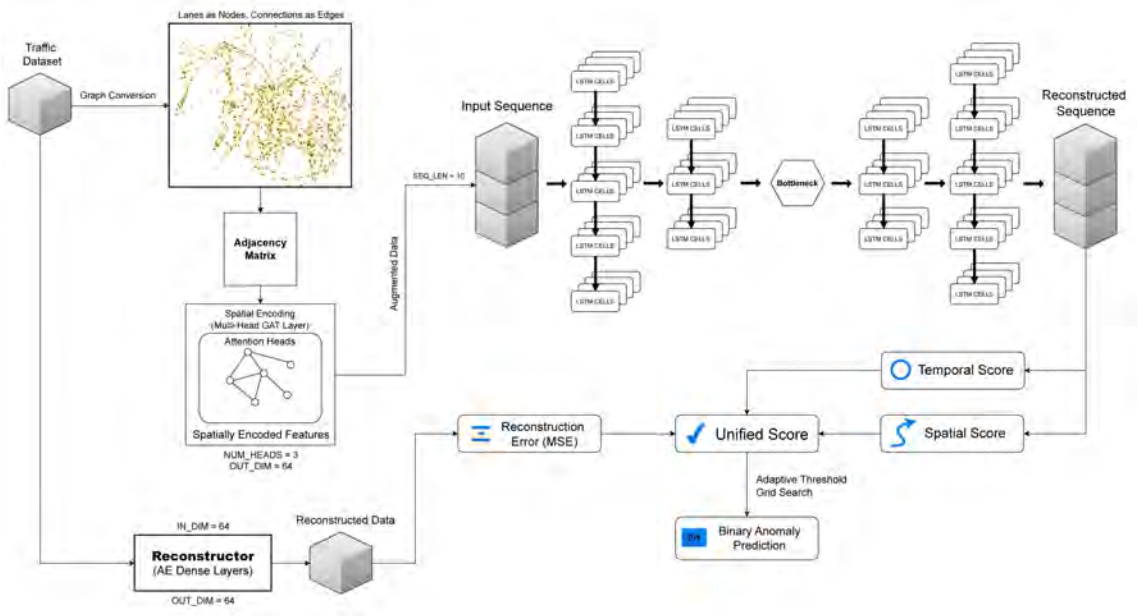

Figure 4.6: Overview of the proposed GATFA architecture.

Figure 4.6 presents an overview of the proposed GATFA (Graph Attention and Temporal Fusion Architecture) fusion framework for lane-level traffic anomaly detection. The diagram illustrates the complete processing flow from raw lane-wise sensor inputs to final anomaly score computation. The pipeline demonstrates how spatial and temporal dependencies are jointly learned to identify irregular traffic patterns efficiently.

4.4.1 Architecture Overview

The proposed **GATFA** architecture conceptualizes the traffic network as a directed graph $G = (V, E)$, where each node $v_i \in V$ corresponds to a unique lane (identified by `lane_id`) and each edge $(v_i, v_j) \in E$ represents connectivity derived from an adjacency matrix loaded from a CSV file. The number of nodes $|V|$ is determined by the unique lanes in the dataset, mapped via a dictionary `lane_to_node`. At each timestamp t, the observation vector $\mathbf{x}_i^t \in \mathbb{R}^F$ encapsulates F multivariate features, including but not limited to vehicle count (`vehicle_count`), mean speed (`mean_speedkmph`), occupancy (`occupancy`), flow (`flow_veh_per_hr`) and density (`density_veh_per_km`). The objective is to learn latent representations that encapsulate spatial correlations among lanes, temporal patterns across historical observations and fidelity to the original data distribution, subsequently fusing anomaly scores derived from deviations in these representations.
GATFA comprises three independent modules and a post-hoc fusion mechanism:

1. **Reconstruction Module (Dense Autoencoder):** A multi-layer feedforward neural network that encodes the current feature vector into a lower-dimensional latent space and decodes it back. This minimizes the mean squared error (MSE) to model the manifold of normal traffic patterns.

2. **Spatial Module (Graph Attention Encoder):** A single-layer, multi-head Graph Attention Network (GAT) that propagates and aggregates features

across neighboring nodes using attention mechanisms which produces spatially contextualized embeddings.

3. **Temporal Module (LSTM Encoder):** A multi-layer unidirectional Long Short-Term Memory (LSTM) network that processes node-specific historical feature sequences to predict the spatial embedding, capturing sequential dependencies.

Each module generates a distinct anomaly score: MSE for reconstruction, Euclidean distance from baseline for spatial and MSE for temporal prediction. These scores are independently normalized using min-max scaling based on training distributions and fused via a weighted sum, with weights and thresholds optimized through grid search to maximize detection performance.

4.4.2 Graph Construction and Feature Representation

The topological structure of the road network is encoded in a static adjacency matrix $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$, where $A_{ij} = 1$ if there is a connection from lane i to lane j and 0 otherwise. This matrix is loaded from `adjacency_csv` and aligned with the sorted unique lanes. No explicit normalization (e.g., row-wise) is applied in the code, as the GAT implementation handles attention normalization internally.
For each timestamp t, the feature matrix $\mathbf{X}^t \in \mathbb{R}^{|V|\times F}$ is constructed by placing $\mathbf{x}_i^t$ at row i for observed lanes, with unobserved lanes zero-filled. Features are preprocessed by dropping missing values, converting to numeric, sorting by timestamp and scaling using a `MinMaxScaler` fitted solely on normal training samples to map values to $[0, 1]$. Temporal sequences are formed for each observation $(\mathbf{x}_i^t, t, i)$ by collecting the previous $SEQ\_LEN - 1$ historical features for node i up to t, padding with the earliest available feature if fewer than SEQ_LEN are present. This results in sequence tensors of shape (N, SEQ_LEN, F), where N is the number of samples.

4.4.3 Spatial Encoding via Graph Attention

Spatial dependencies among lanes in the traffic network are modeled using a memory-efficient, single-layer, multi-head Graph Attention Network (GAT). This approach is particularly suited for capturing adaptive relationships in sparse graphs, such as road networks, where not all lanes are directly connected. By employing attention mechanisms, the GAT dynamically weights the importance of neighboring nodes' features during aggregation that allows the model to focus on relevant spatial interactions while ignoring irrelevant ones. The memory efficiency arises from processing only non-zero edges in the adjacency matrix $\mathbf{A}$, avoiding dense computations that could be prohibitive for large $|V|$.
Given the feature matrix at timestamp t, $\mathbf{X}^t \in \mathbb{R}^{|V|\times F}$, the GAT first applies head-specific linear transformations to project the input features into a higher-dimensional space. For each head $h = 1$ to GAT_NUM_HEADS, the projected features for node i are computed as:

$$\mathbf{z}_i^{(h)} = \mathbf{W}^{(h)}\mathbf{x}_i^t, \tag{4.8}$$

where $\mathbf{W}^{(h)} \in \mathbb{R}^{GAT\_HIDDEN\times F}$ is a learnable weight matrix specific to head h.

Next, attention coefficients are calculated exclusively over the non-zero entries of $\mathbf{A}$ using sparse indexing, which identifies source-destination pairs (i, j) where $A_{ij} > 0$. For each such edge in head h, the unnormalized attention score is:

$$e_{ij}^{(h)} = \text{LeakyReLU}\left(\mathbf{a}_{src}^{(h)} \cdot \mathbf{z}_i^{(h)} + \mathbf{a}_{dst}^{(h)} \cdot \mathbf{z}_j^{(h)}\right), \tag{4.9}$$

with $\mathbf{a}_{src}^{(h)}, \mathbf{a}_{dst}^{(h)} \in \mathbb{R}^{GAT\_HIDDEN}$ being learnable attention vectors for source and destination nodes, respectively. The LeakyReLU activation introduces non-linearity, allowing the model to capture complex dependency patterns.
To ensure numerical stability during softmax normalization the maximum attention score per source node is subtracted. This prevents overflow in exponentiation. The normalized attention coefficients are then:

$$\alpha_{ij}^{(h)} = \frac{\exp(e_{ij}^{(h)} - \max_k e_{ik}^{(h)})}{\sum_k \exp(e_{ik}^{(h)} - \max_k e_{ik}^{(h)}) + \epsilon}, \tag{4.10}$$

where $\epsilon = 10^{-16}$ is a small constant to avoid division by zero in cases of isolated nodes or empty neighborhoods.
These coefficients guide the weighted aggregation of neighbor features for each head:

$$\mathbf{o}_i^{(h)} = \sum_{j \in N(i)} \alpha_{ij}^{(h)} \mathbf{z}_j^{(h)}, \tag{4.11}$$

where $N(i)$ denotes the set of neighbors of node i including potentially self-loops if present in $\mathbf{A}$. $\alpha_{ij}^{(h)}$ is derived from (4.10)
The outputs from all heads are concatenated, passed through an ELU activation for non-linearity and finally projected to the desired output dimension:

$$\mathbf{h}_i^t = \mathbf{W}_{proj}\left(\|_{h=1}^{GAT\_NUM\_HEADS} \text{ELU}(\mathbf{o}_i^{(h)})\right), \tag{4.12}$$

where $\mathbf{W}_{proj} \in \mathbb{R}^{GAT\_OUT\_DIM \times (GAT\_NUM\_HEADS \cdot GAT\_HIDDEN)}$ is a learnable projection matrix. This yields the spatial embedding matrix $\mathbf{h}^t \in \mathbb{R}^{|V| \times GAT\_OUT\_DIM}$, where each $\mathbf{h}_i^t$ encapsulates the spatially contextualized state of node i at time t, informed by its local graph neighborhood.
This process, as detailed in Equations (4.8) through (4.12), enables the model to adaptively learn spatial representations that are robust to varying traffic topologies and density.

4.4.4 Temporal Encoding via LSTM

The temporal module employs a unidirectional Long Short-Term Memory (LSTM) network to model sequential dependencies in the historical features of each node which predicts the spatial embedding derived from the GAT. This prediction task enforces the LSTM to learn temporal patterns, such as trends in speed or occupancy over time, which are crucial for distinguishing normal cyclic behaviors (e.g., rush hours) from anomalous deviations.
The temporal encoding focuses on the last hidden state $\mathbf{o}_i^t[-1] \in \mathbb{R}^{LSTM\_HIDDEN}$, which summarizes the entire sequence. This is linearly projected to align with the GAT output dimension:

$$\hat{\mathbf{h}}_i^t = \mathbf{W}_{fc}\mathbf{o}_i^t[-1] + \mathbf{b}_{fc}, \tag{4.13}$$

where $\mathbf{W}_{fc} \in \mathbb{R}^{GAT\_OUT\_DIM \times LSTM\_HIDDEN}$ and $\mathbf{b}_{fc} \in \mathbb{R}^{GAT\_OUT\_DIM}$ are learnable parameters. The output $\hat{\mathbf{h}}_i^t$ serves as a prediction of the true spatial embedding $\mathbf{h}_i^t$ from Equation (4.12),facilitating the computation of temporal anomaly scores.

4.4.5 Dense Reconstruction

The reconstruction module is implemented as a dense autoencoder that is designed to learn a compact representation of normal traffic features and reconstruct them faithfully. This serves as a baseline for detecting anomalies through reconstruction errors, grounded in the assumption that normal data lies on a lower-dimensional manifold, while anomalies deviate from it.
Operating on the current feature vector $\mathbf{x}_i^t \in \mathbb{R}^{F'}$ (with $F' = F - 2$, excluding auxiliary fields like `lane_node` and `anomaly_type_fact`), the autoencoder architecture comprises an encoder-decoder structure with batch normalization and dropout for regularization. The encoding path compresses the input as follows:

$$\mathbf{l}_1 = \text{ReLU}(\text{BN}(\mathbf{W}_1 \mathbf{x}_i^t + \mathbf{b}_1)), \quad \mathbf{l}_1 \in \mathbb{R}^{64}, \tag{4.14}$$

$$\mathbf{l}_1 = \text{Dropout}(0.2)(\mathbf{l}_1), \tag{4.15}$$

$$\mathbf{l}_2 = \text{ReLU}(\text{BN}(\mathbf{W}_2 \mathbf{l}_1 + \mathbf{b}_2)), \quad \mathbf{l}_2 \in \mathbb{R}^{32}, \tag{4.16}$$

where $\mathbf{W}_1 \in \mathbb{R}^{64 \times F'}$, $\mathbf{W}_2 \in \mathbb{R}^{32 \times 64}$ and associated biases $\mathbf{b}_1, \mathbf{b}_2$. The bottleneck $\mathbf{l}_2$ represents the latent code.
The decoder mirrors this to reconstruct:

$$\mathbf{l}_3 = \text{ReLU}(\text{BN}(\mathbf{W}_3 \mathbf{l}_2 + \mathbf{b}_3)), \quad \mathbf{l}_3 \in \mathbb{R}^{64}, \tag{4.17}$$

$$\hat{\mathbf{x}}_i^t = \sigma(\mathbf{W}_4 \mathbf{l}_3 + \mathbf{b}_4), \quad \hat{\mathbf{x}}_i^t \in \mathbb{R}^{F'}, \tag{4.18}$$

with $\mathbf{W}_3 \in \mathbb{R}^{64 \times 32}$, $\mathbf{W}_4 \in \mathbb{R}^{F' \times 64}$, biases $\mathbf{b}_3, \mathbf{b}_4$ and σ denoting the sigmoid activation to bound outputs in $[0, 1]$, matching the min-max scaled features.
Training minimizes the mean squared error over N normal samples:

$$\mathcal{L}_{rec} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2. \tag{4.19}$$

This objective, combined with the architecture in Equations (4.14)-(4.18), encourages the autoencoder to capture the intrinsic structure of normal traffic data, making high reconstruction errors indicative of anomalies.

4.4.6 Anomaly Score Computation

To detect anomalies, GATFA computes three complementary scores per sample, leveraging deviations in reconstruction, spatial and temporal domains. Baseline spatial embeddings are first established by averaging GAT outputs over normal training timestamps for each node i:

$$\bar{\mathbf{h}}_i = \frac{1}{T_i} \sum_{t \in \mathcal{T}_{train,i}} \mathbf{h}_i^t, \tag{4.20}$$

where $\mathcal{T}_{train,i}$ is the set of training timestamps with observations for node i and $T_i = |\mathcal{T}_{train,i}|$.

For a test sample $(\mathbf{x}_i^t, t, i)$, the reconstruction score quantifies data fidelity:

$$s_r(i,t) = \|\mathbf{x}_i^t - \hat{\mathbf{x}}_i^t\|_2^2, \tag{4.21}$$

as the MSE from the autoencoder output in Equation (4.18).
The spatial score measures deviation from the node's normal embedding:

$$s_s(i,t) = \|\mathbf{h}_i^t - \bar{\mathbf{h}}_i\|_2, \tag{4.22}$$

using the Euclidean norm, where $\mathbf{h}_i^t$ is from Equation (4.12).
The temporal score assesses prediction accuracy:

$$s_t(i,t) = \|\hat{\mathbf{h}}_i^t - \mathbf{h}_i^t\|_2^2, \tag{4.23}$$

as the MSE between the LSTM prediction in Equation (4.13) and the GAT embedding.
These raw scores (Equations (4.21)-(4.23)) are normalized using min-max scaling based on their training distributions:

$$\tilde{s} = \frac{s - \min(s_{train})}{\max(s_{train}) - \min(s_{train}) + 10^{-8}}, \tag{4.24}$$

which ensures comparability across scales and facilitating fusion (here 10^{-8} is used to avoid corner case like zero division).
This multi-faceted scoring, rooted in Equations (4.20) and (4.21)-(4.23), allows GATFA to detect anomalies manifesting in different modalities, enhancing overall robustness.

4.4.7 Score Fusion and Thresholding

The normalized scores are fused into a unified anomaly indicator through a weighted aggregation, with the weight optimized via grid search to balance the contributions of reconstruction against the combined spatio-temporal signals. Specifically, the unified score is:

$$S(i,t) = w_r \tilde{s}_r(i,t) + (1 - w_r)(\tilde{s}_t(i,t) + \tilde{s}_s(i,t)), \tag{4.25}$$

where $\tilde{s}_r, \tilde{s}_s, \tilde{s}_t$ are from Equation (4.24). The optimal w_r maximizes the F1-score on labeled test data only for hyperparameter tuning purposes.
Thresholding employs a percentile-based approach on the unified scores over a grid search to maximize test F1. A sample is classified as anomalous if $S(i,t)$ exceeds the optimized threshold.

4.4.8 Training and Inference Pipeline

The procedure is summarized in Algorithm 2.

Algorithm 2 Training and Inference of GATFA

Require: Graph adjacency $\mathbf{A}$, normal training data $\mathcal{D}_{\text{train}}$, validation $\mathcal{D}_{\text{val}}$, sequence length SEQ_LEN, learning rate η

Initialize dense autoencoder, GAT and LSTM parameters
Train autoencoder on normal features to minimize $\mathcal{L}_{rec}$ (Eq. 4.21 as loss)
for each epoch = 1 to $EPOCHS\_GAT$ **do**
  for each timestamp t in $\mathcal{D}_{\text{train}}$ **do**
    Construct $\mathbf{X}^t$ (zero-fill unobserved)
    Compute $\mathbf{h}^t = \text{GAT}(\mathbf{X}^t, \mathbf{A})$
    For observed nodes, compute $\hat{\mathbf{h}}^t = \text{LSTM}(\mathbf{S}^t)$
    Update parameters to minimize $\mathcal{L}_{temp} = 1 - \frac{1}{M} \sum \cos(\hat{\mathbf{h}}^t, \mathbf{h}^t)$
  end for
  Compute validation loss; save best model
end for
Compute baselines $\bar{\mathbf{h}}_i = \frac{1}{T_i} \sum_{t \in \mathcal{D}_{\text{train},i}} \mathbf{h}_i^t$
for each test sample (i, t) **do**
  Compute s_r, s_s, s_t (Eqs. 4.21-4.23)
  Normalize to $\tilde{s}_r, \tilde{s}_s, \tilde{s}_t$
  Fuse $S = w_r \tilde{s}_r + (1 - w_r)(\tilde{s}_t + \tilde{s}_s)$ with optimal w_r
  Classify as anomaly if $S \geq \tau$ (optimal percentile on train S)
end for

4.4.9 Architectural Advantages

The GATFA architecture offers several key advantages over traditional spatio-temporal anomaly detectors:

1. **Multi-Modal Anomaly Signals:** By combining reconstruction MSE (capturing data fidelity), spatial deviation (highlighting topological inconsistencies) and temporal prediction error (detecting sequential anomalies), GATFA provides comprehensive coverage of diverse anomaly types.

2. **Memory-Efficient GAT Implementation:** The sparse attention computation using edge indices scales efficiently to large graphs which avoids dense matrix operations and enabling handling of extensive lane networks.

3. **Sequence-Based Temporal Modeling:** The LSTM's prediction of spatial embeddings from raw feature histories enforces learning of temporal dynamics with the cosine loss promoting alignment between temporal predictions and spatial contexts.

Overall, GATFA achieves a principled integration of graph-based spatial modeling, sequence-based temporal prediction and reconstruction-based fidelity, ensuring adaptability to diverse traffic scenarios, resilience to data variations and high detection accuracy through mathematical rigor in encoding and scoring.

Chapter 5

Result Analysis

5.1 Performance Evaluation

5.1.1 Evaluation Criteria (Metrics)

F1 Score:
The F1 Score, as defined in Equation (5.3), represents the harmonic mean of Precision (5.1) and Recall (5.2). It is particularly suitable for this task because it penalizes imbalances more heavily than an arithmetic mean which ensures that both metrics are reasonably high for a good F1 Score.
The constituent metrics are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.3}$$

where TP (True Positives) denotes the number of anomalous records correctly identified, FP (False Positives) represents normal records incorrectly flagged as anomalous and FN (False Negatives) indicates anomalous records missed by the model. The F1 Score is particularly valuable in anomaly detection due to the typically skewed distribution of classes where normal instances vastly outnumber anomalies. A high F1 Score indicates that the model achieves a balance between detecting true anomalies and avoiding false alarms.

ROC AUC (Receiver Operating Characteristic - Area Under Curve):
The Receiver Operating Characteristic (ROC) curve and its associated Area Under Curve (ROC-AUC) provide a threshold-independent evaluation of a model's discriminative ability. The ROC curve plots the True Positive Rate (5.4), against the False Positive Rate (5.5) across various decision thresholds. The AUC summarizes the overall performance of the model, with a value of 1.0 indicating perfect separation of anomalous and normal instances and a value of 0.5 suggesting performance no better than random guessing.

The TPR and FPR are defined as:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{5.4}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{5.5}$$

where TN (True Negatives) represents the number of normal records correctly identified as such.
The ROC AUC is particularly advantageous in anomaly detection because rather than relying on a single operating condition, it evaluates the model's ability to rank anomalous instances higher than normal ones. This is especially relevant in scenarios where the optimal threshold is not known or may vary depending on the application. At the same time, The ROC AUC has limitations in highly imbalanced datasets as it gives equal weight to TPR and FPR. So false positives may have a disproportionately large impact due to the abundance of normal instances. In such cases, a model can achieve a high AUC by correctly classifying most negative instances, even if it misses many anomalies. This limitation underscores the need for complementary metrics like the F1 Score and Average Precision.

Average Precision (AP):
Average Precision (AP) offers a threshold-independent evaluation by summarizing the Precision-Recall curve which plots Precision against Recall at various thresholds. Unlike the ROC-AUC which considers both positive and negative classes equally, AP focuses exclusively on the positive class (anomalies), making it particularly suitable for imbalanced datasets where anomalies are rare. Mathematically, it can be approximated as:

$$\text{AP} = \sum_k (\text{Recall}_k - \text{Recall}_{k-1}) \cdot \text{Precision}_k \tag{5.6}$$

where Recall_k and Precision_k are the Recall and Precision values at the k-th threshold.
In anomaly detection, AP is a valuable metric because it emphasizes the model's ability to maintain high Precision while achieving high Recall which is critical when anomalies are sparse. A high AP indicates that the model can reliably identify anomalies without generating excessive false positives. It complements the F1 Score and ROC-AUC by providing a nuanced view of the model's performance across the entire Precision-Recall curve. While the F1 Score is tied to a specific threshold and the ROC AUC considers both classes, AP focuses on the positive class and is less sensitive to the dominance of negative instances. This makes it an essential tool for evaluating models in scenarios where the cost of missing an anomaly is high and the number of normal instances is more than the number of anomalies.

5.2 Result and Analysis

5.2.1 Confusion Matrix

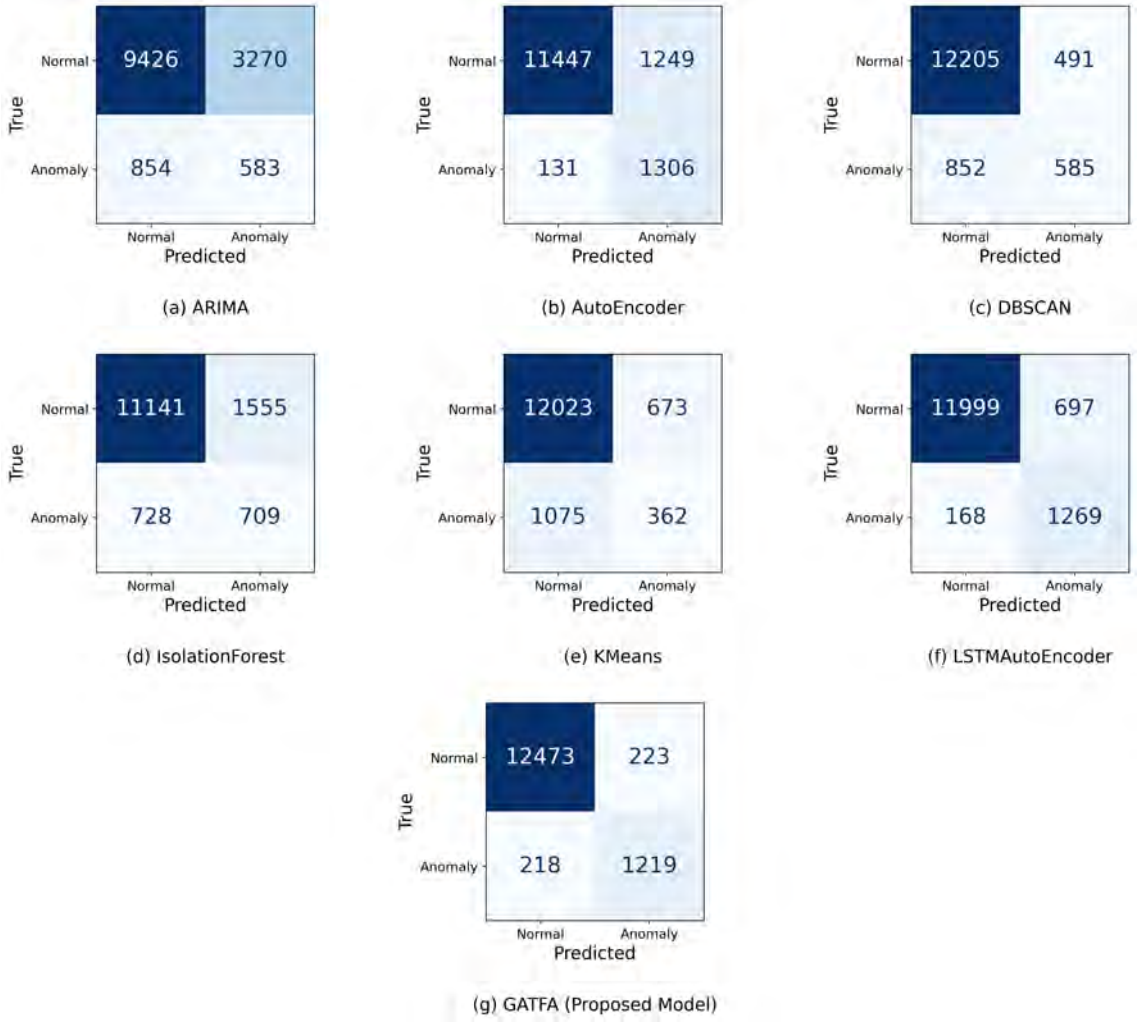

Figure 5.1: Confusion matrices of different anomaly detection models. Subplot (a) corresponds to ARIMA, (b) to AutoEncoder, (c) to DBSCAN, (d) to Isolation Forest, (e) to K Means, (f) to LSTM AutoEncoder, and (g) to the proposed GATFA model. Each subplot shows counts of True Negatives, False Positives, False Negatives and True Positives

The figure 5.1 presents the confusion matrices for seven anomaly detection models: (a) ARIMA, (b) AutoEncoder, (c) DBSCAN, (d) to Isolation Forest, (e) K Means, (f) LSTM AutoEncoder and (g) GATFA. Isolation Forest, DBSCAN and K Means moderately identify normal traffic instances but struggle to detect a significant portion of anomalies. ARIMA performs adequately for normal events but captures very few anomalies due to its reliance on time-series forecasting without spatial context. The AutoEncoder detects a large number of anomalies but also produces a high number of false positives. This results in balanced yet suboptimal performance. The LSTM AutoEncoder improves anomaly detection by leveraging temporal sequences which reducing false negatives compared to non-sequential models. In contrast, the proposed GATFA model achieves a high true positive count (1219) and the lowest

false positive count (223) which demonstrates superior balance in detecting both normal and anomalous instances. This performance stems from GATFA's integration of graph-based spatial features, temporal sequences and feature reconstruction which collectively enhance its ability to capture complex traffic dynamics. Fairness was ensured by applying identical training, validation and test splits across all models. This consistent data partitioning guaranteed that the resulting confusion matrices were directly comparable which allowing for an unbiased assessment of each model's performance.

5.2.2 F1 Score Comparison

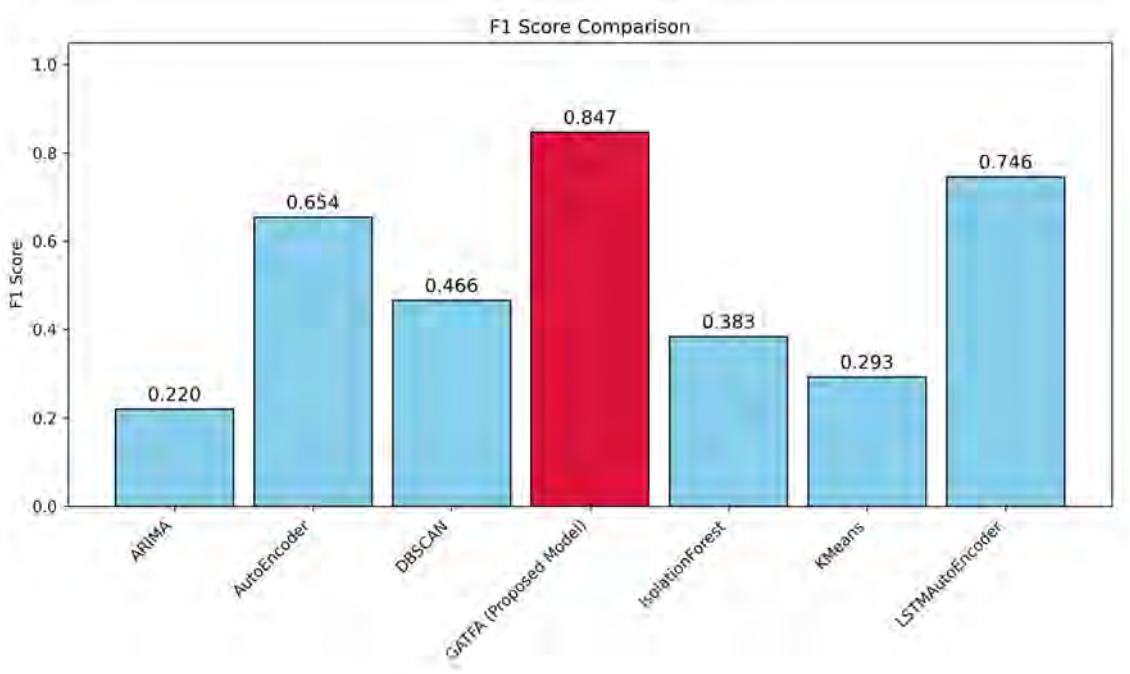

Figure 5.2: Performance Comparison of Unsupervised Models using F1 Scores

The comparative analysis of F1-scores across all evaluated models reveals distinct performance trends that highlight the importance of both temporal and spatial feature learning. As shown in Figure 5.2, the proposed **GATFA model** achieves the highest F1-score of **0.8468**, significantly outperforming all baseline methods. This superior performance indicates the model's strong capability to identify anomalous traffic patterns while maintaining a balanced trade-off between precision and recall. In contrast, traditional models such as the **ARIMA** (0.2204), **Isolation Forest** (0.3831) and **K-Means Clustering** (0.2929) exhibit low F1-scores. Although these models can capture certain underlying trends, they lack the structural adaptability to represent complex nonlinear dependencies in spatio-temporal data. The **AutoEncoder** (0.6543) demonstrates moderate performance, yet its reconstruction-based learning still struggles to distinguish subtle deviations that arise due to contextual dependencies between nodes while Isolation Forest's random partitioning often misclassifies dense anomalous regions. Similarly, ARIMA's reliance on linear temporal assumptions limits its robustness when facing abrupt or irregular fluctuations in traffic flow. Clustering-based approaches, including **DBSCAN** (0.4656)

and **K-Means Clustering** (0.2929), continue to yield the lowest F1-scores among the machine learning methods tested. Their poor performance can be attributed to their static feature representation and inability to model time-dependent patterns efficiently. On the other hand, the **LSTMAutoEncoder model** achieves a comparatively high F1-score of **0.7458**, demonstrating a significant advantage of integrating temporal sequence modeling with an autoencoding structure. By capturing long-term dependencies, this model effectively detects evolving anomalies that persist over time. Still performance remains lower than the proposed GATFA model primarily because it focuses predominantly on temporal features without fully accounting for fine-grained spatial interdependencies among connected road segments. The GATFA model overcomes this limitation by integrating graph attention mechanisms with temporal fusion and achieves a more holistic understanding of the data which results in a higher F1 performance.

5.2.3 ROC-AUC Comparison

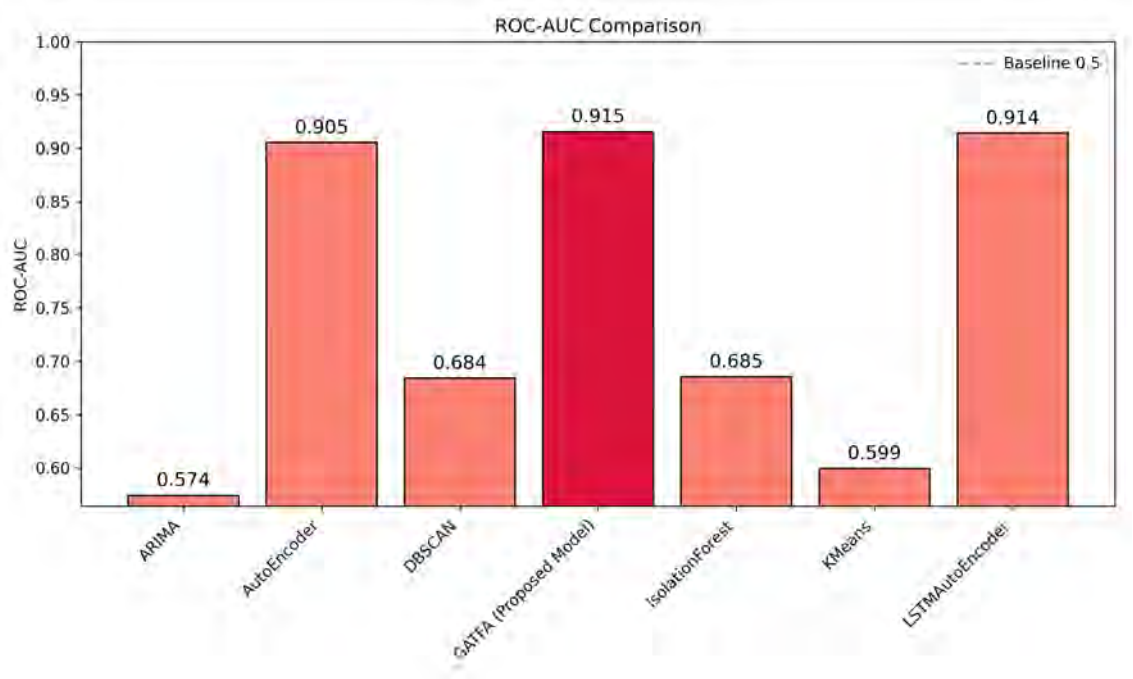

Figure 5.3: Performance Comparison of Unsupervised Models using ROC-AUC value

The ROC-AUC analysis as illustrated in Figure 5.3 provides additional evidence of the superior discriminative capability of the proposed **GATFA model**. Achieving the highest ROC-AUC value of **0.9154**, the model demonstrates exceptional proficiency in distinguishing between normal and anomalous traffic states across varying threshold levels. This result indicates a strong balance between true positive and false positive rates ultimately confirming that GATFA generalizes effectively to complex traffic behaviors. Among the baseline methods, the **AutoEncoder** (0.9052) and **LSTMAutoEncoder** (0.9141) achieve relatively higher ROC-AUC scores which their partial ability to capture nonlinear and sequential dependencies within the data. The AutoEncoder benefits from its latent-space representation of normal behavior while LSTMAutoEncoder leverages temporal dynamics to detect

persistent anomalies over time. However, both models are limited by their inability to jointly model spatial interactions among correlated road segments, resulting in reduced robustness compared to GATFA. In contrast, traditional and clustering-based algorithms such as the **IsolationForest** (0.6855), **DBSCAN** (0.6842), **KMeans** (0.5995) and **ARIMA** (0.5741) exhibit noticeably weaker discriminative performance. These methods rely on simplified assumptions like statistical isolation, distance-based clustering, or linear temporal forecasting that fail to capture the intricate spatio-temporal dependencies inherent in real-world traffic systems. Consequently, their decision boundaries are less adaptive to the dynamic and correlated nature of anomaly patterns. Overall, the ROC-AUC results reaffirm that integrating *graph-based spatial attention* with *temporal sequence modeling* provides a more comprehensive understanding of traffic dynamics. The proposed GATFA framework effectively leverages these complementary features which results in a more robust and generalizable anomaly detection model.

5.2.4 Average Precision Comparison

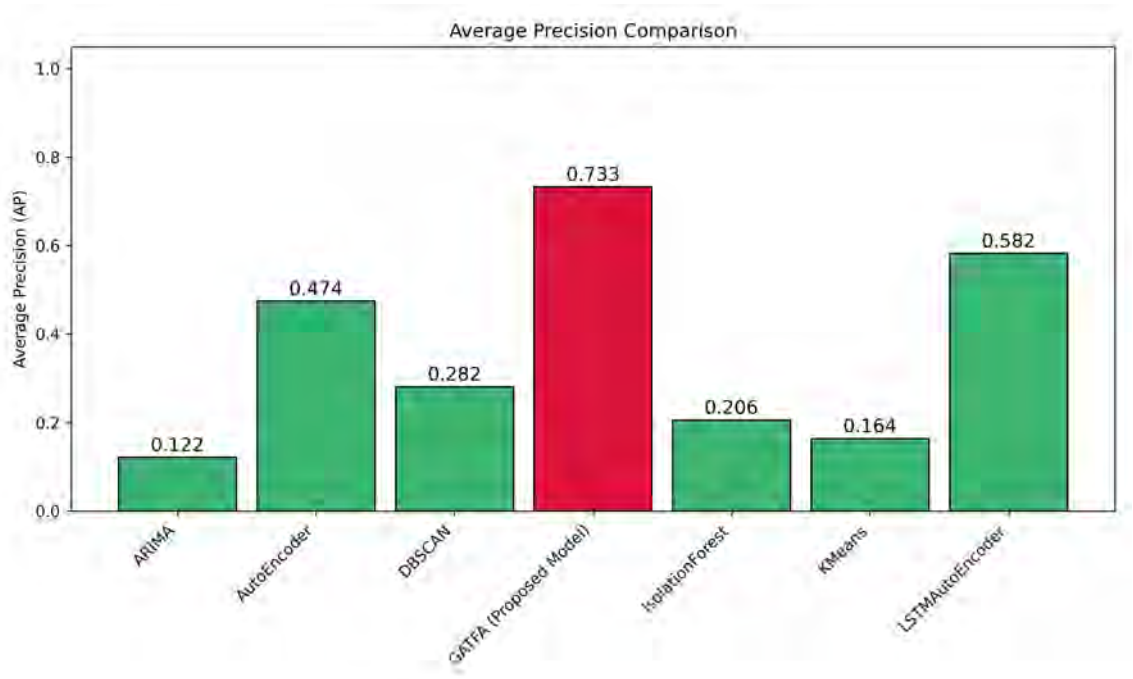

Figure 5.4: Performance Comparison of Unsupervised Models using Average Precision value

As presented in Figure 5.4, our proposed GATFA model achieves the highest average precision score of 0.733, marking a significant improvement over the next best baseline, LSTMAutoEncoder which attains an AP of 0.582. This substantial performance gap underscores GATFA's superior capability in identifying anomalies within highly imbalanced datasets. In contrast,traditional methods such as ARIMA and KMeans exhibit significantly lower performance. These methods fail to capture the nuanced patterns required for effective anomaly detection in such settings. Similarly, IsolationForest and DBSCAN show modest performance, likely due to their

reliance on density-based or isolation-based heuristics that are less effective when class distributions are heavily skewed. AutoEncoder performs better than these traditional methods but falls short of LSTMAutoEncoder and GATFA, as it lacks the specialized mechanisms to handle temporal or relational data effectively.

5.2.5 PR Curve

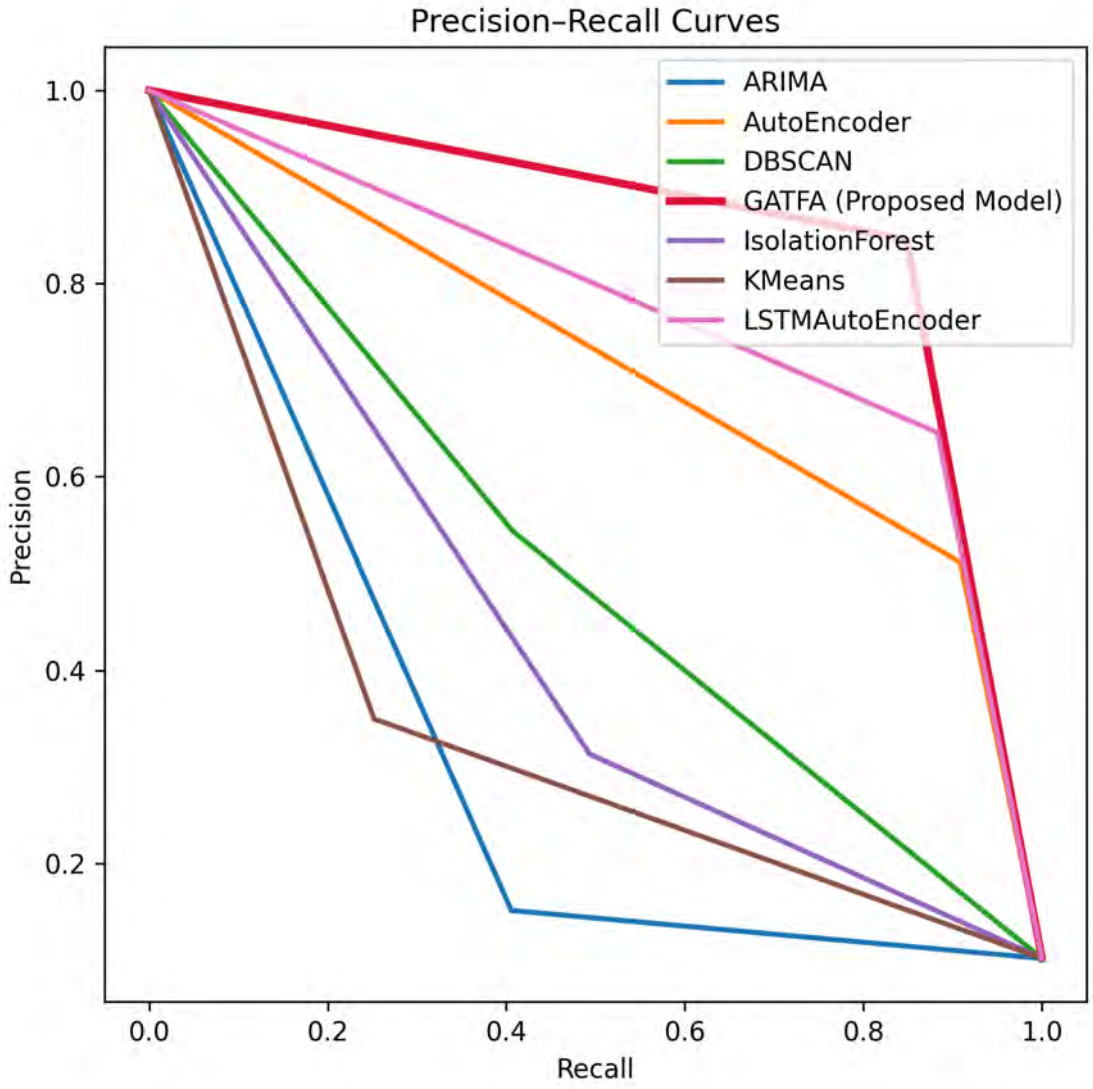

Figure 5.5: Precision–Recall Curves of Unsupervised Models highlighting performance under class imbalance

The Precision–Recall (PR) curves shown in Figure 5.5 provide a comprehensive evaluation of model performance in the context of class imbalance which is a common challenge in real-world anomaly detection tasks. These curves quantify the trade-off between precision and recall for each unsupervised model and focus exclusively on their binary anomaly detection capabilities to ensure a fair and consistent comparison. To achieve this, the PR curves are constructed by evaluating each model's ability to classify instances as either anomalous or normal with decision

thresholds varied to capture the full spectrum of precision–recall trade-offs. The resulting staircase-like appearance of the curves is an expected artifact of the binary classification approach and it aligns with standard practices for evaluating anomaly detection performance. Since the evaluated models detect anomalies using different mechanisms such as reconstruction errors, graph-based attention scores, clustering distances etc., plotting PR curves against raw errors or model-specific scores would be unfair and could bias the comparison. By thresholding each model's outputs into binary predictions, we maintain a consistent and equitable basis for evaluation across all methods.

5.2.6 ROC Curve

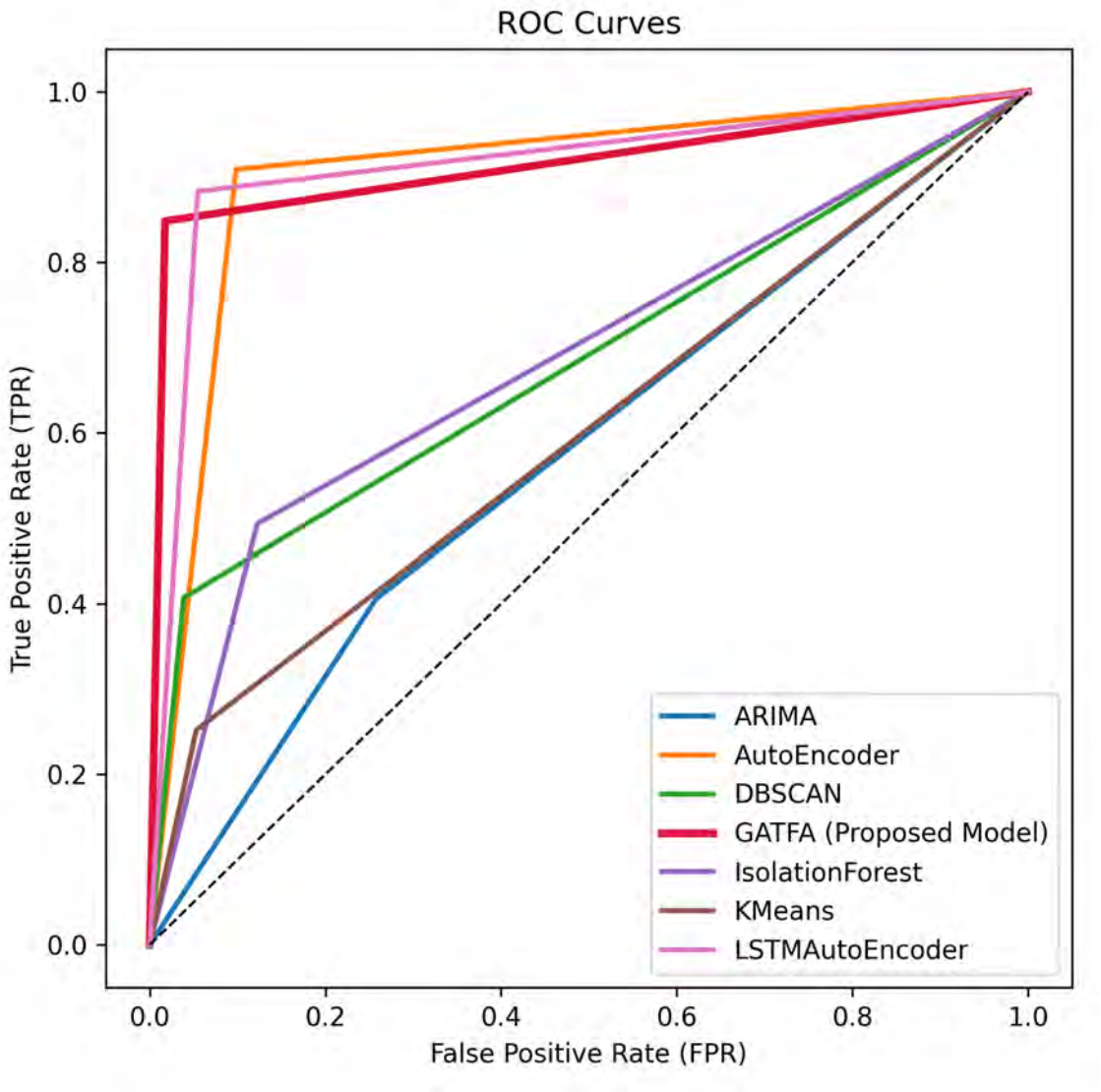

Figure 5.6: ROC AUC Curves of different unsupervised models showing trade-offs between True Positive Rate and False Positive Rate

The **ROC-AUC curves** in Figure 5.6 provide a comparative evaluation of several unsupervised anomaly detection models, including **AutoEncoder**, **Isolation Forest**, **DBSCAN**, **K-Means**, **LSTM**, **ARIMA** and the proposed **GATFA** model.

Each curve depicts the trade-off between the *True Positive Rate (TPR)* and the *False Positive Rate (FPR)* across varying threshold values which highlights each model's ability to distinguish anomalous from normal traffic. Curves that approach the top-left corner indicate stronger detection performance, reflecting higher TPRs with minimal FPRs. To ensure fairness, all models were evaluated on identical dataset splits which allows a consistent comparison of through ROC curves. Since the models detect anomalies using different mechanisms, directly plotting ROC curves against raw model-specific scores would be misleading. By converting outputs to binary predictions for ROC-AUC computation, the evaluation provides an fair basis for comparing models with fundamentally different scoring approaches. The staircase-like appearance of these curves is also an expected artifact of thresholding binary predictions in anomaly detection tasks. Among the evaluated methods, **GATFA** achieves the highest AUC, demonstrating its ability to capture complex spatio-temporal dependencies in lane-level traffic data, while **LSTM** and **AutoEncoder** also perform competitively, achieving AUC-ROC values significantly close to GATFA. In contrast, classical statistical models like **ARIMA** and clustering-based approaches such as **K-Means** show lower discriminative power due to their limited capacity to adapt to dynamic spatial correlations. Overall, the ROC-AUC analysis confirms that graph-based deep architectures, particularly GATFA, provide more reliable and consistent anomaly detection across diverse traffic conditions, emphasizing the benefit of combining graph attention mechanisms with temporal feature aggregation.

5.3 Analysis of the Design Solution

The GATFA architecture showcases several strengths that contribute to its effectiveness in spatio-temporal anomaly detection. By combining graph attention mechanisms with recurrent modeling,the framework captures complex spatial dependencies and temporal dynamics inherent in large-scale traffic networks.The attention mechanism allows the model to adaptively weigh the influence of neighboring nodes that enables it to distinguish between local fluctuations and system-wide anomalies. The integration of a LSTM further enhances temporal awareness which is essential for understanding evolving traffic patterns. Moreover, the hybrid reconstruction-based anomaly scoring method, augmented with temporal smoothing, enhances robustness against noise and allows the model to reliably detect diverse anomaly types, including point, contextual and collective irregularities. The use of an end-to-end learning structure also promotes generalization without requiring handcrafted features or explicit subgraph partitioning.

Despite its advantages, the GATFA design also presents several challenges and limitations. The architectural complexity introduced by multi-head attention layers, recurrent modules and decoder networks increases the number of hyperparameters and the computational demand during both training and inference. Threshold calibration for anomaly scoring remains sensitive and can influence the model's balance between precision and recall, particularly in dynamic traffic conditions. While the model demonstrates strong representational capacity, real-time deployment in large-scale systems may be constrained by latency and memory requirements. Achieving efficient inference would therefore require further optimization through techniques

such as pruning, quantization or adaptive temporal windowing. In summary, while GATFA offers a powerful and unified approach for modeling spatio-temporal traffic behavior, its operational efficiency and parameter sensitivity necessitate careful consideration for practical implementation.

5.4 Final Design Adjustments

Following end-to-end evaluation of the GAT-LSTM Autoencoder pipeline, several adaptive refinements were applied to stabilize training, balance loss components and improve anomaly detection reliability:

1. **If reconstruction loss stagnates or diverges:** increase encoder–decoder depth or hidden dimension, apply dropout regularization on input embeddings and tune the learning rate scheduler (e.g., reduce LR decay rate or warmup duration).

2. **If validation loss remains high while training loss decreases:** introduce early stopping or L2 weight decay and enable denoising regularization (dropout or noise injection on node embeddings) to prevent overfitting.

3. **If false positives dominate anomaly predictions:** recalibrate the detection threshold on the validation set by maximizing F1 or controlling the false-positive rate and apply temporal smoothing on per-lane anomaly scores (e.g., persistence filtering where anomalies must persist for $> k$ timesteps).

4. **If training is unstable across epochs:** monitor epoch-wise metrics (train loss, validation loss, learning rate) and use gradient clipping or smaller batch size. Check printed epoch summaries to confirm convergence patterns.

5. **If GPU memory or training time is constrained:** reduce GAT attention heads or lower hidden dimensions. Optionally use GraphSAINT or neighbor-sampling variants to optimize per-batch graph construction.

6. **If lane-wise signals exhibit high noise:** apply robust aggregation using median or trimmed mean across sensors and assign reliability weights to lanes based on historical variance in their readings.

5.4.1 Limitations

- SUMO-simulated anomalies might not capture all real-world sensor noise or human-driven edge cases; transfer to field data may require domain adaptation.

- Hyperparameter sensitivity and complexity of combining multiple loss terms require rigorous validation.

Chapter 6

Conclusion

6.1 Summary of Findings

The experimental results demonstrate that the proposed **GATFA** model significantly outperformed the six baseline models: AutoEncoder, Isolation Forest, DBSCAN, K-Means, LSTM and ARIMA. While the baseline models showed varying levels of effectiveness in detecting anomalies, their performance was limited by factors such as sensitivity to parameter tuning, inability to capture complex spatial-temporal dependencies and reduced robustness under diverse traffic conditions. In contrast, the GATFA leveraged its dual spatial-temporal graph structure and cross-fusion mechanism to achieve superior accuracy, precision, recall and F1-score. These findings highlight the effectiveness of integrating spatial-temporal graph learning with fusion techniques which confirms that the proposed model is better suited for real-time traffic anomaly detection in dynamic and heterogeneous urban environments.

6.2 Contributions to the Field

This research contributes to the field of intelligent transportation systems by advancing anomaly detection methodologies through the integration of graph-based modeling and self-supervised learning. Unlike traditional grid-based and purely statistical approaches, our proposed GATFA framework captures both spatial and temporal dependencies in dynamic traffic networks while maintaining scalability for large urban environments. This work creates a thorough baseline for traffic anomaly identification and prepares the road for realistic AIoT-driven smart city implementations by methodically assessing six unsupervised models before moving on to a novel graph attention network based architecture. By doing this, it not only solves the current issues with data sparsity, scalability, and accuracy, but it also lays a strong basis for further study into resilient and adaptive traffic management systems.

6.3 Recommendations for Future Work

Looking ahead, the proposed traffic anomaly detection framework can evolve into a far more intelligent, interactive and inclusive system for modern urban mobility. One major step forward would be the creation of an intuitive user interface that

allows traffic authorities to visualize live conditions, track anomalies as they occur and interact with detailed dashboards that reveal historical patterns or emerging risks. Integrating augmented reality (AR) tools could even allow field operators to visualize congestion zones or accidents in real time. Beyond visualization, the system can incorporate advanced anomaly classification models that not only detect but also categorize incidents such as accidents, congestion, or roadblocks and assess their severity. By learning from past events, these models could prioritize urgent situations and automatically suggest timely interventions. Scalability and robustness are also critical for real-world deployment. Expanding the system to support multiple cities or larger networks will require optimized performance and resilience against missing or noisy data. Predictive modeling could take this further by anticipating potential disruptions before they happen which allows authorities to proactively prevent congestion or accidents through smart rerouting strategies. The future also lies in connecting this system with broader smart city infrastructures. Linking it to emergency services, public transportation and weather forecasting will enable truly coordinated urban mobility management. Leveraging edge computing will ensure low-latency,real-time processing,making the system faster and more efficient even in high-traffic scenarios. Finally, involving city officials,traffic operators and everyday commuters through continuous feedback will make the platform more adaptive, user-centered and aligned with real-world needs that will transform it into a cornerstone of intelligent transportation in the coming years.

In an era of rapid urban growth and increasingly complex transportation demands, this study presents a real-time lane-level traffic anomaly detection framework. It actively enhances the intelligence and responsiveness of vehicle-road collaborative systems. By integrating Graph Attention Networks with temporal fusion encoding, the model continuously learns spatio-temporal dependencies within dynamic traffic flows, allowing it to identify irregularities with precision and minimal manual supervision. The self-supervised mechanism and attention-driven architecture optimize computational efficiency while preserving high detection accuracy. As cities continue to expand, this framework is shaping a new generation of adaptive, data-driven traffic management systems that promote safer roads, reduced congestion, and more sustainable urban mobility.

References

[1] J. Ji et al., *Spatio-temporal self-supervised learning for traffic flow prediction*, arXiv preprint, 2022. [Online]. Available: https://arxiv.org/abs/2212.04475

[2] T. Huang, C. Liu, A. Sharma, and S. Sarkar, "Traffic system anomaly detection using spatiotemporal pattern networks," *International Journal of Prognostics and Health Management*, vol. 9, no. 1, 2020. DOI: 10.36001/ijphm.2018.v9i1.2671 [Online]. Available: https://doi.org/10.36001/ijphm.2018.v9i1.2671

[3] C. Ai, L. Jia, M. Hong, and C. Zhang, "Short-term road speed forecasting based on hybrid rbf neural network with the aid of fuzzy system-based techniques in urban traffic flow," *IEEE Access*, vol. 8, pp. 69 461–69 470, 2020. DOI: 10.1109/ACCESS.2020.2986278 [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2986278

[4] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2378–2392, 2021. DOI: 10.1109/tnnls.2021.3068344 [Online]. Available: https://doi.org/10.1109/tnnls.2021.3068344

[5] J. Bharadiya, "Artificial intelligence in transportation systems: A critical review," *American Journal of Computing and Engineering*, vol. 6, no. 1, pp. 34–45, 2023. DOI: 10.47672/ajce.1487 [Online]. Available: https://doi.org/10.47672/ajce.1487

[6] A. Ajayi and H. Kumkale, "Optimising urban road transportation efficiency: Ai-driven solutions for reducing traffic congestion in big cities," 2023, Preprint. [Online]. Available: https://doi.org/10.13140/RG.2.2.16130.45763

[7] A. A. Ouallane, A. Bahnasse, A. Bakali, and M. Talea, "Overview of road traffic management solutions based on iot and ai," *Procedia Computer Science*, vol. 198, pp. 518–523, 2022. DOI: 10.1016/j.procs.2021.12.279 [Online]. Available: https://doi.org/10.1016/j.procs.2021.12.279

[8] S. K. Jagatheesaperumal, S. E. Bibri, J. Huang, J. Rajapandian, and B. Parthiban, "Artificial intelligence of things for smart cities: Advanced solutions for enhancing transportation safety," *Computational Urban Science*, vol. 4, no. 1, 2024. DOI: 10.1007/s43762-024-00120-6 [Online]. Available: https://doi.org/10.1007/s43762-024-00120-6

[9] K. K. Santhosh, D. P. Dogra, and P. P. Roy, "Anomaly detection in road traffic using visual surveillance," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–26, 2020. DOI: 10.1145/3417989 [Online]. Available: https://doi.org/10.1145/3417989

[10] A. Barodi, A. Bajit, T. E. Harrouti, A. Tamtaoui, and M. Benbrahim, "An enhanced artificial intelligence-based approach applied to vehicular traffic signs detection and road safety enhancement," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 1, pp. 672–683, 2021.

[11] J. Zhao, X. Yang, and C. Zhang, "Vehicle trajectory reconstruction for intersections: An integrated wavelet transform and savitzky-golay filter approach," *Transportmetrica A: Transport Science*, 2023. DOI: 10.1080/23249935.2022.2163207 [Online]. Available: https://doi.org/10.1080/23249935.2022.2163207

[12] H. Wu, "The internet-of-vehicle traffic condition system developed by artificial intelligence of things," *Journal of Supercomputing*, vol. 78, pp. 2665–2680, 2022. DOI: 10.1007/s11227-021-03969-0 [Online]. Available: https://doi.org/10.1007/s11227-021-03969-0

[13] Q. Li, S. Meng, S. Wang, J. Zhang, and J. Hou, "Cad: Command-level anomaly detection for vehicle-road collaborative charging network," *IEEE Access*, vol. 7, pp. 34 910–34 924, 2019. DOI: 10.1109/ACCESS.2019.2904047 [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2904047

[14] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, 2020. DOI: 10.1109/ACCESS.2020.2988796 [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2988796

[15] U. Rashid, M. F. Saleem, S. Rasool, A. Abdullah, H. Mustafa, and A. Iqbal, "Anomaly detection using clustering (k-means with dbscan) and smo," *Journal of Computing & Biomedical Informatics*, vol. 7, no. 2, 2024. DOI: 10.56979/702/2024 [Online]. Available: https://doi.org/10.56979/702/2024

[16] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1479–1489, 2021. DOI: 10.1109/TKDE.2019.2947676 [Online]. Available: https://doi.org/10.1109/TKDE.2019.2947676

[17] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*, Halifax, NS, Canada, 2017, pp. 665–674.

[18] B. Liu, X. Tang, J. Cheng, and P. Shi, *Traffic flow combination forecasting method based on improved lstm and arima*, College of Information Science & Technology, Hainan University & University of Chinese Academy of Science, 2019.

[19] Z. Lu, W. Lv, X. Zhipu, B. Du, and R. Huang, "Leveraging graph neural network with lstm for traffic speed prediction," Aug. 2019, pp. 74–81. DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00056

[20] S. F. Ahmed et al., "Enhancement of traffic forecasting through graph neural network-based information fusion techniques," *Information Fusion*, vol. 110, p. 102 466, 2024, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2024.102466 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253524002446

[21] T. Wu, F. Chen, and Y. Wan, "Graph attention lstm network: A new model for traffic flow forecasting," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, 2018, pp. 241–245. DOI: 10.1109/ICISCE.2018.00058

[22] F. Caetano, P. Carvalho, and J. Cardoso, "Deep anomaly detection for in-vehicle monitoring—an application-oriented review," *Applied Sciences*, vol. 12, no. 19, p. 10 011, 2022. DOI: 10.3390/app121910011 [Online]. Available: https://doi.org/10.3390/app121910011

[23] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018. arXiv: 1710.10903 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1710.10903

[24] Y. Zhang et al., "A cooperative vehicle-road system for anomaly detection on vehicle tracks with augmented intelligence of things," English, *IEEE Internet of Things Journal*, vol. 11, no. 22, pp. 35 975–35 988, 2024, Publisher Copyright: © 2014 IEEE., ISSN: 2327-4662. DOI: 10.1109/JIOT.2024.3398023