Scientific Reports

https://doi.org/10.1038/s41598-025-33193-2

Article in Press

Hybrid deep learning techniques for adaptive routing and congestion control in urban VANET for wireless mobile networking

Received: 14 August 2025

Accepted: 16 December 2025

Published online: 03 April 2026

Cite this article as: Maray M. Hybrid deep learning techniques for adaptive routing and congestion control in urban VANET for wireless mobile networking. *Sci Rep* (2026). https://doi.org/10.1038/s41598-025-33193-2

Mohammed Maray

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

ARTICLE IN PRESS

© The Author(s) 2026. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

Hybrid Deep Learning Techniques for Adaptive Routing and Congestion Control in Urban VANET for Wireless Mobile Networking

Mohammed Maray[1,*]

[1]Department of Informatics and Computer Systems, College of Computer Science, King Khalid University, Abha, Saudi Arabia. Mmarey@kku.edu.sa

[2]Central Labs, King Khalid University, AlQura'a, Abha, P.O. Box 960, Saudi Arabia.

**** Corresponding author: Mohammed Maray; Mmarey@kku.edu.sa***

Abstract

The dynamic nature of traffic, high mobility and varying topology are major challenges to Urban Vehicular Ad Hoc Networks (VANETs), especially in densely populated urban areas. The methods of traditional routing and congestion control are not able to adjust and this leads to increased latency, decreased throughput and frequent breakdowns. In this paper, a new hybrid deep learning architecture that combines Graph Convolutional Networks (GCN) to extract spatial features and Long Short-Term Memory (LSTM) networks to model time is proposed with the aim of offering adaptive routing and other proactive congestion control capabilities in urban VANETs. The model applies the real time traffic information of sensors and vehicular communication systems in urban areas to forecast traffic congestion, model vehicle density and route optimization. In experimental results, the proposed GCN-LSTM model is significantly more efficient in routing and also in predicting results, compared to the baseline models. In particular, it recorded a Mean Absolute Error of 0.02, Root Mean Squared Error (RMSE) of 0.07, routing latency of 38.13 ms and Packet Delivery Ratio of 95, which was much better than GCN-only (MAE: 4.65, PDR: 90.3) and CNN-LSTM (MAE: 0.14, PDR: 88.1). The hybrid model also aids the real-time traffic processing through the edge-cloud cooperation, reducing the inference latency and scaling to city regions. Such methods as geospatial embedding and temporal batching enable the model to represent the intricate flow of traffic in high-resolution. The architecture is also modular and allows simple incorporation of both emergency vehicle prioritization and location-based rerouting and is scalable to operate on resource-constrained edge nodes, meaning it can be used in practice in intelligent city infrastructure. Altogether, the developed model offers a high-performing and scalable adaptive routing and congestion control solution to smart urban transportation systems.

Keywords: *VANET, Adaptive Routing, Congestion Control, GCN, LSTM, Urban Mobility, Wireless Networking, Deep Learning and Traffic Forecasting.*

1. Introduction

The urbanization and growth of mobile computing have led to high demand of efficient wireless network routing in the heavily populated cities [1]. The necessity of smooth communication between mobile devices, vehicles, and infrastructure in cities is becoming a crucial issue as they become smarter to achieve reliability, low latency, and congestion-free communication [2]. Mobile Computer Networks (MCNs), Vehicular Ad-Hoc Networks, and Urban Air Mobility systems are becoming more and more based on real-time information-driven routing to work best in dynamic and congested urban environments [3]. The main issues that these networks encounter are the high rate of topology changes, interference of signals and the density of nodes [4]. The quality of service (QoS) is also worsened by congestion in the urban road networks and communication link congestion, leading to packet loss, delay increase and reduced throughput [5]. To overcome these problems, adaptive routing frameworks that learn spatial and time time trends are critical in reducing congestion[6]. Deep learning techniques have been shown to be successful in tackling on complex spatiotemporal decisions in urban networked systems[7].

However, existing solutions for routing and congestion control often struggle in rapidly changing urban environments. [8] proposed a cloud-integrated AI–IoT framework using Random Forest for real-time E. Coli prediction with 99% accuracy through LoRaWAN networks. This concept informs the proposed GCN–LSTM model by inspiring cloud-assisted, AI-driven adaptive routing and congestion prediction in urban VANET systems. Reinforcement Learning-based techniques such as Q-routing and Deep Q-Networks have been employed to learn the best paths by interacting with dynamic environments [9]. These models, however, suffer from high convergence times and instability in fast-changing urban topologies. Convolutional Neural Networks and classical Feedforward Neural Networks have been used to predict traffic congestion but fail to capture spatial dependencies among mobile nodes and the underlying network graph [10]. Additionally, traditional models such as Support Vector Machines and Random Forests lack the scalability and flexibility required for real-time routing decisions [11]. These approaches are either too superficial to represent spatiotemporal dynamics or too inflexible to learn network dynamics effectively [12].

Graph Neural Networks proved capable of learning topological relation among entities within networks and anticipating routing measures [13]. Although GNNs have succeeded in modeling spatial dependencies well, they lack adequacy in characterizing traffic status evolution in terms of time in highly mobile networks such as those in VANETs [14]. LSTM networks perform adequately in forecasting a long time domain but do not possess the feature of knowing spatial arrangement of structures such as road maps as well as connections between nodes [15]. They attempt to integrate these elements but lean so much on graph structures with fixed forms or cannot manage real-time variations in congestion [16]. They also tend to ignore important aspects such as queue length, signal levels or dynamic buffer status diminishing their use in realistic urban network [17].

To address the limitations of existing methods, this study proposes a novel hybrid deep learning framework that integrates Graph Neural Networks and Long Short-Term Memory networks. Proposed architecture leverages GNN to capture spatial relationships among mobile nodes in urban wireless graphs and LSTM to model temporal dynamics of congestion and traffic patterns. As compared to existing solutions, the proposed system supports sophisticated feature extraction methods with node velocity, queue length, signal quality and traffic density.

The proposed model outperforms existing traditional methods, with respect to predictive accuracy, routing efficiency, and overall QoS in congested cities. Over the last few years, the interaction between Information and Communication Technology (ICT) and smart devices has allowed for a more flexible and intelligent approach to routing in the urban environment. Connected vehicles, roadway sensors, and roadside units (RSUs) serve as smart devices that greatly assist in supporting real-time decision making to aid effective traffic management. Since these devices are dynamic in nature, they transmit real-time traffic data that can be used by smart routing systems to both enhance traffic flow, reduce congestion, and enhance the overall performance of the Vehicular Ad-Hoc Network. Smart routing is distinctive as it not only reacts to the state of traffic conditions, but also takes into account other conditions including weather, road closings, and priorities of vehicles related to emergencies, etc., making it a valuable aspect of next generation smart city infrastructure.

The proposed hybrid deep learning architecture combines these real-time, distributed data sources to adaptive routing, which is consistent with the increasing demand to have smart cities which use intelligent transportation system to facilitate their mobility. The suggested hybrid deep learning model exhibits some of the major developments in adaptive routing and congestion control of urban VANETs. It is built to scale smoothly in urban settings effectively managing networks of thousands of vehicles and several traffic sensors and handling massive amounts of real-time traffic information. The real-time adaptive routing feature of this work is one of its main accomplishments since it enables vehicles to dynamically modify their paths in response to real-time congestion predictions through optimization of the traffic flow and reduction of delays in dense urban environments. Moreover, edge-cloud collaboration enables the increase of the system scalability and the decrease of inference latency by transferring heavy computations to the cloud so that in congested conditions, the response time is very high. As opposed to other traditional routing approaches, such as Dijkstra and Bellman-Ford algorithms, which do not take into consideration dynamic traffic conditions, the hybrid GCN-LSTM model is always better in terms of lowering the latency and increasing the percentage of packet delivery and predicting congestion more accurately.

Scope of the Paper

The given paper is devoted to the method of enhancing the adaptive routing and congestion control in urban Vehicular Ad-Hoc Networks and suggests a hybrid deep learning architecture, which integrates both the Graph Convolutional Networks that extract the spatial features and the Long Short-Term Memory that performs a temporal prediction. The boundaries of this research are narrowed down to urban VANETs, in which a high velocity of vehicles, existing road dynamics, and different traffic flows are challenging the traditional routing schemes. This study will improve routing speed and real-time prediction of congestion in actual urban traffic scenarios which could offer a scalable and adaptable solution to smart city infrastructures.

Open Issues in Data Dissemination

Notwithstanding the improvements in VANETs, there are a number of gaps in data dissemination that are still left unsolved especially in city settings. It is a challenging task because the vehicle-infrastructure real-time data synchronization is highly mobile, the topology can be easily changed, and traffic congestion greatly impacts real-time data. Moreover, scalability is also an important issue, with the growth in the number of vehicles within a city, and the network resources get overloaded. The existing VANET communication infrastructures are usually not able to effectively process the amount of information produced by sensors, vehicles, and infrastructure, which results in delays and data losses. Additionally, the data security, network congestion, and the feasibility of wireless communication in the city are issues that are not completely covered by current models. As much as this paper will provide a solution to adaptive routing and congestion control, these wider issues of distributing data in real time urban VANETs will be subject to future research.

Contributions

This contributes the following in the first place: the proposed hybrid deep learning architecture, to be integrated into urban VANETs to enhance adaptive routing and congestion management, is a novel type of deep learning architecture that combines both the Graph Convolutional Networks and Long Short-Term Memory networks to extract spatial and temporal features respectively.

- The creation of an actual time traffic congestion prediction system that links traffic flow information, automobile density, and network conditions, performing dramatic enhancements on the efficacy of routing in addition to the accuracy of the congestion forecasting.

- A comparative performance study that can illustrate the effectiveness of the proposed GCN-LSTM model over the available models on the key performance indicators including Mean Absolute Error, Root Mean Squared Error, routing latency, and Packet Delivery Ratio.

Contribution and Novelty of Work:

The given work presents an innovative hybrid deep learning architecture that encompasses both the Graph Convolutional Networks (GCN) and Long Short-Term Memory (LSTM) networks to perform adaptive routing and congestion control in urban VANETs (Vehicular Ad Hoc Networks). The key contributions and novelties of this study are:

Combination of GCN and LSTM:

The combination of Graph Convolutional Network (GCN) with spatial feature extraction and Long Short-Term Memory (LSTM) networks with a temporal model is an innovative solution that can successfully cope with both spatial and temporal complexities of dynamic urban VANETs. This combination enables the model to adjust to the dynamism in the urban traffic patterns besides being able to model the intrinsic topological association in the vehicular network.

RTAR, Proactive LTCM:

The model suggested takes advantage of real-time traffic sensors of the city and vehicle communication networks to estimate traffic congestion, vehicle density modeling, and dynamically optimize routes. This work stands out of the conventional methods of congestion management since it offers proactive methods of managing congestion, unlike the traditional approach that responds to the network performance only after the problem of congestion has been experienced.

Increased Scalability and Real-Time:

The model employs complex feature engineering such as temporal batching, statistical encoding, and geospatial embedding which facilitate the model to learn complex spatiotemporal dynamics. The techniques not only improve the prediction accuracy and routing efficiency of the complex traffic behaviour in urban environment, but also make the model more effective in addressing the complex traffic behaviour.

The improved accuracy through the application of advanced feature engineering:

The results of the experiment show that the GCN-LSTM hybrid model is much higher in terms of the Mean Absolute Error, the Root Mean Squared Error, the routing latency, and the Packet Delivery Ratio than its baseline models, i.e., GCN-only and CNN-LSTM. Specifically, the proposed model determines a PDR (Packet Delivery Ratio) of 95 per cent that is considerably higher than the approaches that could be used in the previous years, which highlights its effectiveness in ensuring consistent communication in urban VANETs.

Higher Performance than Base models:

The findings of the experiment indicate that the GCN-LSTM hybrid model has a significant higher performance in regard to Mean Absolute Error, Root Mean Squared Error, routing latency, and Packet Delivery Ratio compared to the baseline models, i.e., GCN-only and CNN-LSTM. In particular, the model suggested identifies a PDR (Packet Delivery Ratio) of 95 per cent which is notably greater than the methods available in the past, which underscores its strength in providing a stable flow of communication within urban VANETs.

Future Improvement Modular Architecture:

The proposed framework is also modular and therefore can be easily integrated with other functionalities such as emergency vehicle prioritization and location-based rerouting that lead to flexibility as far as adjusting it to other purposes of smart cities is concerned.

1.1 Problem Statement

Vehicular ad-hoc networks (VANETs) and intelligent city infrastructure are examples of urban wireless mobile computer networks that are critically struggling to ensure the efficient routing and congestion control [18]. The highly mobile urban environment of strong dynamics is making it hard to optimally carry out routing and attain real-time behavior due to topological changes that take place with high frequency and network conditions that vary [19]. The existing strategies are inclined to strike an incorrect balance in minimizing latency, maximizing throughput and avoiding congestion in these highly interconnected and complicated networks. It becomes an issue in terms of sustaining a stable communication and movement of vehicles [20].

Adaptive routing the Q-routing RL-based methods have been applied to solve adaptive routing. However, these approaches have slow convergence rates and require a lot of computation, hence not efficient in real-time processing of dynamic urban networks [21]. Though they eventually converge to optimal solutions, exploration and exploitation typically take a long time particularly in large-scale networks where there is a need for continuous learning [22]. Exploration and exploitation are lengthy processes though they eventually settle on the best solutions especially in large scale systems whereby continuous learning is required [22].

Machine learning-based traffic prediction models utilizing methods such as Support Vector Machines or decision trees have been used to predict congestion [23]. Though they are overly simplistic and make use of fixed features and don't capture intricate, non-linear interdependence between nodes within a network in an urban setup [24]. They tend to overlook temporal and spatial traffic condition variations which cause erroneous congestion predictions and ineffective routing. These models lack sufficient response to highly dynamic urban traffic patterns and network states [25].

Dijkstra's and Bellman-Ford are among traditional routing algorithms most commonly applied to determine the shortest path in network graphs [26]. However, these algorithms fail to perform well in real-time congestion in cities where road blockages, accidents or traffic explosion suddenly change the topology of the network [27]. These methods fail to take into consideration network congestion and the time dependent delays which are highly valued in smart routing decisions in smart cities. Such algorithms are not able to utilize real time traffic data to forecast and avoid congestion by restricting the performance during highly dynamic and congested urban systems [28].

Despite the significant advances in routing mechanisms and traffic prediction technologies, the existing models cannot reflect the complexity of urban wireless networks [29]. Problems that remain unsolved include inefficient adaptation, computational time and a lack of congestion and network dynamics that can be incorporated. What is required is a more comprehensive, dynamic model capable of predicting and controlling congestion in a dynamic manner as well as optimizing the real-time routing decisions and offering scalability to the urban conditions [30].

1.2 Objective

- Develop hybrid deep learning framework combining Graph Convolutional Networks and Long Short-Term Memory for adaptive routing and traffic congestion forecasting in urban VANETs.
- Study the spatial and temporal patterns of urban vehicle traffic from high-resolution mobility and communication records.

- Apply a feature engineering workflow consisting of temporal batching, statistical encoding, queue monitoring, and geospatial embedding to augment model input.
- Assess performance of suggested hybrid model based on significant metrics like MAE (Mean Absolute Error), RMSE, routing latency and Packet Delivery Ratio (PDR).
- Compare suggested GCN-LSTM model with baseline models such as single GCN and CNN-LSTM based on prediction accuracy and scalability.

1.3 Structure of Paper

Section 1: Introduction emphasizes motivation, issues in urban VANET environments and shortcomings in current routing and congestion control methods. Section 2: Literature Survey provides a detailed review of relevant works such as CNNs, LSTMs, GRUs, Transformers, GCNs, and hybrid models applied to traffic forecasting and vehicular routing. Section 3: Suggested Hybrid DL for Adaptive Routing and Congestion Management in Urban VANET describes the GCN-LSTM framework's architecture. Further, it is subdivided into major subsections: 3.1 Data Collection, 3.2 Preprocessing Layer, 3.3 Feature Extraction, 3.4 Hybrid GCN and LSTM Model, 3.5 Routing & Congestion Layer, and 3.6 Performance Assessment, each providing a description of the technical parts and mechanisms associated with the model pipeline. Section 4: Results and Discussion compares the performance with GCN and CNN-LSTM models by evaluating model through visualizations and metrics such as MAE, packet delivery ratio, routing latency and RMSE. Section 5: Conclusion and Future Enhancement concludes results, confirms model's superiority. This organization provides logical flow from problem identification to the confirmation of the proposed solution.

2. Literature Survey

By and large, in the past couple of decades, improvements in deep learning have tremendously increased traffic prediction and routing in urban vehicular networks. Xing et al. [31] designed a CNN-based model to extract spatial features from vehicle trajectories and road sensor maps, which was trained on GPS data of urban taxis. Their centralized routing engine performed better in accuracy when compared to statistical models; nevertheless, real-time implementation of the model was ensured by reducing convolutional filters and setting up edge servers. Zhai et al. [32] considered LSTMs to capture sequential dependencies in traffic flow with respect to historical and live data. In dynamic environments, their model performed better than RNN and ARIMA, hence providing QoS benefits in proactive rerouting. Abdullah et al. [33] proposed a GRU-based approach to predict congestion points with less computational cost and energy efficiency than LSTM. The authors had their attention-based GRU approach tried out on a VANET simulation testbed, with its reduction in delay and packet loss shown. For traffic prediction, Tedjopurnomo et al. [34] considered Transformer architecture with self-attention, consequently allowing for efficient training and stabilizing context awareness. The model was integrated into a VANET simulator for demonstrating improvements in throughput and reductions in delay.

Graph-based and hybrid models also seem to give promising results. Munuhwa et al. [35] used GCN with the aim of modelling road network graphs to learn spatial correlations between intersections and road segments. This decentralized edge deployment coupled with LSTM for temporal awareness fostered an increase in routing accuracy, especially under dynamically changing conditions. Rui et al. [36] applied Deep Belief Networks for congestion forecasting in delay-tolerant networks thus managing effectively sparse and incomplete data. Kempinska et al. [37] used Variational Autoencoders for unsupervised anomaly detection where reconstruction error was used to detect congestion and reroute traffic. Yan et al. [38] built a Deep Q-Network for adaptive routing where the learning was based on traffic density and signal quality, and they deployed this through a cloud-edge setup for real-time inferencing. Zhang et al. [39] favoured Temporal Convolutional

Networks for long-range traffic forecasting because they were better in generalization and training speed compared to LSTM and GRU. On the other hand, Mahmoud et al. [40] presented a hybrid Bi-LSTM-CNN model that would capture spatial and temporal patterns to foster routing and load balancing in dense urban areas that were robust to missing data.

[41] VANETs for mobile linguistic laboratories emphasize privacy, data integrity, and authentication, employing techniques such as symmetric key cryptography, trust-based node profiling, and privacy-preserving cryptography. While research has addressed the cybersecurity risks like eavesdropping, spoofing, and DoS attacks in dynamic environments, few studies have focused specifically on the lightweight security requirements of philological fieldwork with resource-constrained mobile nodes. New frameworks have enhanced data integrity and operational performance, and a specific framework to match the endangered language documentation is unexplored.

[42] The creation of VANETs based on 5G has enhanced the real-time services such as music streaming though issues such as lost packets and large latency still exist. Network slicing (NS) and predictive beamforming solutions are some of the solutions that can mitigate these problems as they can be used to optimize the latency and resource allocation. Neural networks such as LSTM can be used to forecast the vehicle paths to improve the signal consistency. This study suggests a delay sensitive architecture of NS and predictive beamforming to make 5G VANET music transmission efficient in reducing the latency, PDR and throughput and enhancing it.[43] The growing requirement of HD music streaming in VANETs puts the pressure on the conventional broadcasting protocols as the issue requires low latency and high data rates. Recent solutions combine machine learning and optimization methods, including the DTS-CatBoost model, that forecasts network congestion and optimizes routing using the stable broadcast node selection. There is also forward error correction (FEC) to enhance the reliability of data when there is high mobility. It is demonstrated in experimental work that this method allows improving the packet delivery ratio (PDR), decreasing the latency, and offers seamless HD music streaming, which demonstrates the potential of AI-supported streaming solutions in the next-generation transportation systems.

2.1 Challenges in Solving the Problem

Dynamic and High Mobility Environment: The dynamic nature of cars in an urban environment may make environments extremely dynamic such that it is difficult to predict the congestion patterns and make real-time modifications in routing decisions. Real-Time Data Processing: The difficulties refer to the processing of large real-time data consisting of many sources (sensors, vehicles, and roadside units) to guarantee the high speed of its processing, its precision, and timeliness when giving predictions.

- Scalability: As the number of vehicles within the network grows, the complexity of the operations traffic patterns and the choices of routes also grows. The model should be highly scalable to handle the increasing number of vehicles and road segments.

- Energy Efficiency: The model should be optimized in order to operate on edge devices with a low level of computational power but high accuracy and low latency, particularly in real time applications.

- Data Integration: The combination of data of different sources like traffic sensors, GPS and communication networks is only achievable with advanced preprocessing methods and real-time data synchronization.

3. Proposed Hybrid DL for Adaptive Routing and Congestion Control in Urban VANET

The proposed system architecture in Vehicular Ad Hoc Networks (VANETs) in Figure 1 is supposed to support intelligent adaptive routing and congestion control under the urban environment using hybrid deep learning method. System consists of mobile vehicles in which the wireless communication units are onboard to form a highly dynamic and decentralized ad hoc network. They are able to communicate with each other (V2V) and also with fixed Roadside Units (RSUs) in the short-range wireless communication protocols and facilitate real-time exchange of information related to mobility, traffic intensity and road conditions. RSUs are intermediaries that combine traffic data of multiple vehicles and transmit that data to edge computing or cloud computing servers. The core of the use of hybrid Graph Convolutional Network and Long Short-Term Memory (LSTM) on cloud/edge server is the backbone of intelligence. GCN manages the space relationship and dependencies of the vehicle communication network and LSTM assembles temporal statistics of traffic with time. When used together the models predict the best paths to use when routing and identify possible locations of congestion where traffic can be actively controlled and rerouted to avoid congestion. It enables the ability to manage traffic on an urban scale in real time and at scale and with data to provide vehicles with low communication overhead and quality of service (QoS). With edge computing, there is no latency and the model prediction and routing update can be swiftly pushed back to the RSUs to reach the vehicles. This closed loop system is not only more efficient in routing, it is also capable of dynamically changing with changing traffic characteristics, which makes it an excellent solution to intelligent transportation systems of the future.

The exchange of real-time information between the infrastructure and vehicles is improved with the integration of smart devices (traffic sensors, vehicles with GPS capabilities, roadside units (RSUs)). This partnership combined with sophisticated deep learning algorithms allows adaptive routing systems to make sound decisions using real-time traffic information. The edge-cloud model of collaboration applied in this framework is such that timely predictions of traffic and congestion control are made with the least amount of latency making use of the distributed characteristics of smart devices. The system is able to dynamically adapt to the dynamic features of traffic by integrating the Graph Convolutional Networks (GCN) to extract spatial features, and Long Short-Term Memory (LSTM) networks to forecast congestion, the system has the ability to dynamically schedule the routes as congestion changes occur in real-time.

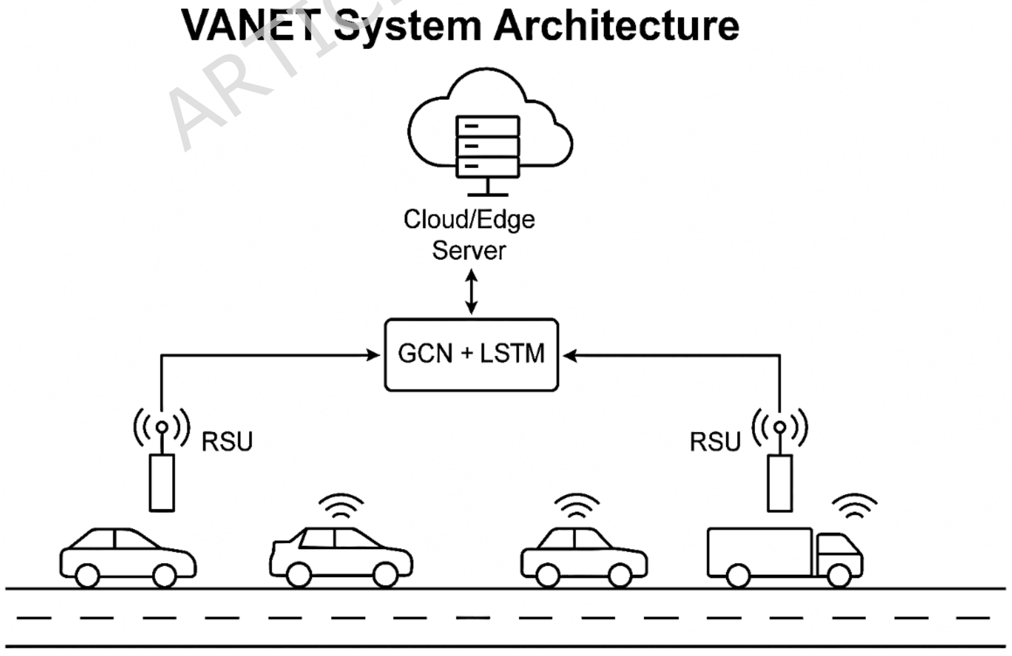

Figure 1: VANET System Architecture with GCN-LSTM-Based Routing and Congestion Control

Dense system architecture specifically for smart routing and traffic congestion management in city area Vehicular Ad Hoc Networks is described in Figure 2. Decentralized vehicle-to-vehicle communication, emergency vehicle-to-emergency vehicle, real-time outdoor perception using roadside traffic monitoring sensors, and robust edge-cloud smarts through hybrid deep learning engines are integrated. Every vehicle interacts with adjacent Roadside Units (RSUs), which act as edge nodes gateways to handle pre-processing, aggregation and local inference of traffic information. Edge nodes function as the first instance of intelligent decision-making, conserving latency and bandwidth usage and facilitating instant congestion detection and rerouting features.

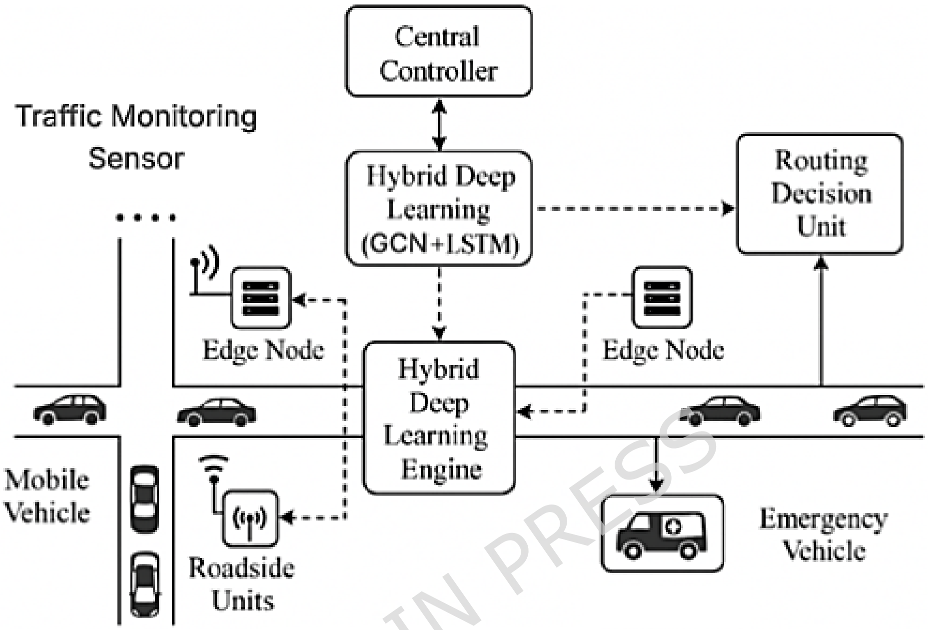

Figure 2: Intelligent VANET Framework Integrating Deep Learning for Real-Time Urban Traffic Optimization

The base intelligence of the system is dedicated to the Hybrid Deep Learning Engine, a combination of the Graph Convolutional Networks (GCNs) to learn spatial relationships between the road segments and clusters of vehicles and Long Short-Term Memory (LSTM) networks to learn time-dependent traffic patterns. The architecture also focuses on one Hybrid Deep Learning (GCN+LSTM) module which is specifically adapted to serve both the engine and the Central Controller as the coordination node in updating the model, recognizing anomalies and police making. By separating the modular learning engine from the primary controller, scalable deployment and policy-guided routing decision are enabled with the flexibility supported in heterogeneous urban environments.

Routing decisions are also dynamically controlled by the Routing Decision Unit which is provided with context-sensitive predictions from the learning engine and considers road constraints, emergency vehicle priority and volatility in traffic. The unit coordinates traffic flows by providing routing instructions to the mobile nodes within VANET via edge infrastructure. Emergency vehicles are provided with special routing support, ensuring improved public safety operations in terms of high congestion. The design supports bi-directional data exchange, in which real-time sensor input, predictive model output, and control feedback loop interact seamlessly to realize proactive and adaptive urban traffic optimization. Layered structure and GCN-LSTM integration support this system to run at scale with responsiveness as well as predictive foresight essential for smart city transportation networks.

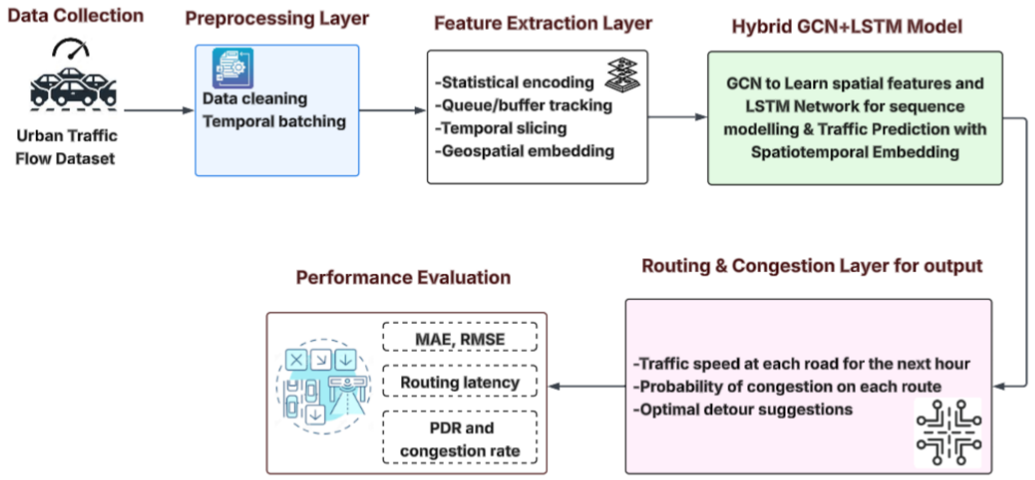

Figure 3: Hybrid GCN+LSTM-Based Urban Traffic Prediction and Routing Pipeline

Hybrid GCN+LSTM Based Urban Traffic Forecasting and Routing Pipeline in Figure 3 describes end-to-end workflow of the deep learning model proposed for adaptive routing and congestion management. Process starts from gathering of urban traffic flow data which goes through preprocessing operations like cleaning data and temporal batching to normalize time-series inputs. In the feature extraction layer, significant features like queue lengths, temporal patterns and geospatial coordinates are represented through statistical and spatial-temporal methods. Encoded features are subsequently input into hybrid GCN+LSTM model, where Graph Convolutional Network (GCN) learns spatial relations between road segments and LSTM handles sequential temporal patterns for precise traffic forecasting. Output layer conducts routing and congestion analysis by predicting traffic speeds, congestion probability forecasting and optimal detour recommendations. Performance assessment is conducted through measures such as MAE, RMSE, routing latency, packet delivery ratio (PDR) and congestion rate giving quantitative evidence of the effectiveness of the model in real-world urban traffic conditions.

Justification for Model Selection

The hybrid Graph Convolutional Network (GCN) and Long Short-Term Memory (LSTM) model has been chosen due to its peculiarity to deal with both challenges of a spatial dependence and time dynamics of the urban Vehicular Ad-Hoc Networks (VANETs). The rationale behind adopting such models is the nature of the particular needs of the urban VANET environments where effective control of traffic jams and adaptive routing are necessitated in highly dynamic environments.

1. Spatial Modeling with GCN

The urban VANETs are associated with the complicated road networks whose conditions of traffic are dynamic and the spatial dependencies of various road segments, vehicles, and intersections have to be reflected to maximize the routing decisions. Graph Convolutional Networks (GCNs) would be exceptionally suitable to this task since it is tailored to work with data organized into graphs. A road network may be modeled in VANETs as a graph according to which intersecting road segments or road crossings can be defined as nodes, and the roads as edges. GCNs are very effective in preserving these spatial relationships allowing the model to know how the traffic conditions at one intersection or road section are impacting others in nearby situations. This plays a vital role in the adaptive routing in the city where the traffic dynamics are changing fast.

2. Temporal Modeling with LSTM

Together with the spatial, there is time-varying traffic congestion, whereby the rush hours, accidents, or road closures add considerable time changes. Long Short-term Memory (LSTM) networks are specifically designed to learn long-term dependencies in time-series data, which is why they can be considered optimal in the context of future traffic forecasting based on the past patterns. LSTMs can be trained in VANETs to predict the effect of congestion at time step t on the future congestion levels at time step t + i with the help of congestion at time step t + j, which are fundamental to routing and congestion control. This property of LSTMs to be able to remember and forget information with time makes sure that the model is able to cope with the changing nature of the traffic data unlike simpler models like the traditional RNNs.

3. Hybrid GCN-LSTM Model

As the GCNs and LSTMs are combined into a hybrid model, a more comprehensive approach of adaptive routing and congestion control is made possible. The GCN is a spatial part of the believers who know the interaction between the different components of the road network, and the LSTM is the time dynamics part of the interactions, the time-dependent evolution of traffic congestion. The importance of this hybrid structure to ensure maximum exploitation of real-time routing decisions is that it combines spatial knowledge (the interaction of the diverse intersections or roads with one another) and temporal knowledge (the variation of the traffic pattern as a function of time). Combining these two effective deep learning methods, they will allow the model to have a more clear and dynamic image of the current traffic condition and the following traffic streams, which will allow them to resort to proactive factors of congestion management and reaction to the decision-making regarding the routing decisions.

Spatio-Temporal Graph Convolutional Network (STGCN) and GraphSAGE-LSTM are conventional graph-based temporal models that have been commonly applied in traffic and mobility prediction because they can capture spatial and temporal correlations. Nevertheless, those models commonly use the static graph structure and have a large computation cost at the training stage, which does not make them very applicable to real-time tasks in dynamic vehicular networks. The proposed hybrid GCNLSTM model, on the other hand, uses a dynamic and lightweight topology that changes the connectivity of nodes with the constantly changing vehicle mobility patterns. This design significantly reduces model complexity while maintaining accurate temporal learning through the LSTM component. As a result, the proposed framework achieves faster inference and better scalability, providing a more practical solution for **resource-constrained VANET environments** without compromising prediction accuracy.

4. Advantages Over Alternative Approaches

The GCN-LSTM hybrid model has a number of specific benefits over alternative approaches, in particular, CNNs or more traditional reinforcement learning (RL) mechanisms, such as Q-routing and Deep Q-Networks: This is because CNNs are useful in feature extraction but constrained in their capability of capturing the spatial relation among various road segments of a network. They do not consider the topological character of urban road networks and are unable to model the interactions between vehicles and infrastructure complexly and spatially dependent.

- Deep RL Methods: Each of the methods such as Q-routing or DQN can learn optimal routes but has high convergence times and poor stability under dynamic conditions such as the urban VANET. Such approaches have a tendency to miss timely choices in urban topology in rapid change, leading to suboptimal routing and delays. Hybrid GCN-LSTM model, however, offers more real-time and correct predictions as it learns spatial and time aspects of traffic.

- Classical Machine Learning: Architectures such as Support Vector Mortar (SVM) or the Random Forests are not scalable or as flexible as needed by real-time decision-making in highly dynamic systems. Such models cannot usually deal with the non-linear nonstationary nature of urban traffic patterns. The hybrid G CNN- LSTM is however developed to support the complex spatiotemporal data, and hence it is able to evolve under different conditions which cannot be achieved by the traditional machine learning methods.

3.1 Data Collection and Dataset Splitting

The data came out of Urban Traffic Flow Dataset available in Kaggle which accumulates the real traffic data of roadways and crossroads in the city. Data measurement was done through assistance of sensors, GPS, and road equipment by advising time stamped vehicular movement, speed, density and congestion rate. It also uses network variables such as the communicational delay, the queue size and parameters of node mobility that are obtained using the vehicular communication networks and the urban wireless networks. The spatiotemporal resolution was high in the data gathering process to enable proper tracing of traffic behavior in different times of the day as well as geographical locations. Such extensive data collection can be used to build deep learning models that are specifically applied to urban VANET adaptive routing and traffic congestion control. The urban traffic flow dataset used in this study, Urban Traffic Flow Dataset, was gathered via a combination of both GPS sensors and roadside units (RSU) installed in different intersections in urban environments. The sensors were able to capture real time traffic information including vehicle density, speed, and GPS position, and were also able to capture time changes in terms of time. Data collection took a duration of 6 months during peak and off-peak hours in order to have diversity in the data. The data was divided into 70% training, 15% validation, and 15 percent testing. The model was developed using the training data and hyperparameter tuning was done using the validation set. The final performance of the model was tested using the test set which was kept absolutely apart. This will ensure the model has been tested on unknown data, and it will not be overfitted to give a realistic assessment of its overall performance in generalization.

3.2 Preprocessing Layer

This layer is the hub of raw network and traffic data preparation to train a deep learning model. It performs important transformations such as numerical feature normalization (e.g. speed, number of lanes and latency), discrete variable encoding (e.g. intersection IDs and type of road) and time-stamped records transformed into structured time formats such as hour of day or day of week. Spatial features such as GPS coordinates are also mapped onto a grid or zone based structure to correspond to spatial attention processes of model. Preprocessing pipeline ensures the consistency of features scales, missing values are interpolated using time-series prediction methods and multivariate sequences completed. This pre-processing improves the quality of data and facilitates effective learning by hybrid deep learning algorithms. These preprocessing methods played an important role in data quality, but it was necessary to determine their effect on prediction accuracy. The Mean Absolute Error (MAE) of the model is 0.02, and the Root Mean Squared error (RMSE) is 0.07, which proves that the effect of missing GPS data, non-uniform sampling, and latency noise did not have a significant impact on the accuracy of the prediction. The model also managed imperfect data, which is manifested by its high level of performance when compared to baseline models. This indicates that the adopted preprocessing techniques were effective in ensuring high quality input data to the deep learning framework. Raw data was pre-treated to take care of the missing values based on forward-fill, and interpolation. Z-score normalization was applied to features that are numeric like vehicle speed, number of flows and latency. The information has been classified into time steps (e.g. every hour), and spatial items were mapped with grid-based coordinates to be combined with the Graph Convolutional Network.

Handling of Missing GPS Data, Irregular Sampling, and Latency Noise

Throughout the data pretreatment phase, several challenges concerning missing GPS data, irregular sampling, and latency noise, arose.

- Missing GPS Data: In instances which the data were missing because of sensor malfunction or data loss, interpolation methods, including forward fill and linear interpolation, were used to create an estimated value of the missing GPS data based on the surrounding data points. This method of interpolation helped with the flow of spatial data while controlling for possible biases in the data.

- Irregular sampling: Given that the data were sampled at various time intervals based on how data were collected through traffic monitoring, temporal batching was used to split the data into consistent, fixed length time windows. Temporal batching allowed the model to be trained on data consistently sampled while accounting for irregular time intervals. Time-stamped GPS coordinates were allocated to a grid-based system to help in the consistency of the data being sampled at different, irregular time intervals.

- Latency noise: Because of the latency that often may occur in vehicular communication, in order to reduce additional noise in the GPS coordinates and latency data, generalized Kalman filtering and smoothing algorithms were applied. Kalman filtering and smoothing algorithms improved the spatial tracking and reduced noise, which increased the stability of the predictions made by the model.

3.2.1 Data Cleaning

Data cleaning process began with the identification and treatment of missing, inconsistent and duplicate records in raw dataset. Missing values particularly for time sequences such as vehicle flow or speed were imputed via forward-fill or interpolation methods in order to preserve continuity of traffic patterns. Statistical methods like Z-score and interquartile range were applied to find outliers and either corrected or removed to avoid biased model training. Duplicate GPS records and logically inconsistent timestamp records were removed to ensure temporal consistency of the sequences. Non-operational sensor or faulty node data were removed to guarantee input feature reliability. Cleaning continued in the form of synchronizing traffic and network layers by synchronizing time windows and location tags among various data flows.

$$x_t = x_{t-1} + \frac{(x_{t+1} - x_{t-1})}{(t+1-(t-1))} = \frac{x_{t-1} + x_{t+1}}{2} \tag{1}$$

Where:

- X_t represents the state or value of the system at time step t (e.g., traffic flow, vehicle speed, etc.).
- X_{t-1} represents the state at the previous time step, $t-1$.
- X_{t+1} represents the state at the next time step, $t+1$.
- The term $\frac{(X_{t+1} - X_{t-1})}{(t+1-(t-1))}$ computes the change in the system between two-time steps ($t-1$ and $t+1$) normalized by the difference in time steps (which is 2, since $(t+1) - (t-1) = 2$).
- The equation $\frac{X_{t-1} + X_{t+1}}{2}$ computes the average of the values at the previous time step X_{t-1} and the next time step X_{t+1}, which can be useful for smoothing or approximating intermediate values.

For example, network latency information needed to be synchronized with matching vehicle movement records to produce valid feature sets for congestion forecasting. GPS noise was minimized through smoothing algorithms like Kalman filtering to enable precise spatial tracking. Categorical data such as intersection IDs or road types were standardized to a similar format to be consistent. Competent cleaning procedure removed redundancy and distortion and made processed data input into the deep learning scheme model real-world traffic movement with high fidelity. The time-series data were temporal batched into fixed-size sequential windows that are fed into deep learning models.

$$\hat{x}_k = \hat{x}_{k-1} + K_k(z_k - \hat{x}_{k-1}) \quad (2)$$

Where $\hat{x}_k$ estimated position, z_k observed noisy measurement and K_k Kalman gain.

3.2.2 Temporal Batching

A batch represents a particular time interval and contains sequential data points for all applicable features like vehicle flow, speed and network latency. Through these temporal batches, the model is allowed to learn dependencies and patterns over time, like the accumulation and dissipation of traffic congestion. Overlapped sliding windows were employed during batching to maintain temporal continuity and enhance the model's capacity for predicting near-future events based on recent patterns. This temporal batching technique not only facilitates short-term congestion prediction but also allows adaptive routing decisions in dynamic VANETs. $T_k = \{x_i | t_i \in [t_k, t_k + \Delta t]\}$ denote batch for time window k.

$$v_i^{norm} = \frac{v_i - \mu_v}{\sigma_v} \quad (3)$$

where μ_v and σ_v are mean and standard deviation of speed.

In every batch, the accuracy of the mapping between traffic features and network variables is guaranteed so that the model may examine vehicular movements as a function of wireless communications states over time. Each batch contains GPS coordinates that are time-stamped to preserve spatial-temporal context, useful for adequately routing cars across continent-spanning road networks. Organizing data in this manner improves compute efficiency and gives the models input in a manner useful to learn complex sequential dependencies.

3.3 Feature Extraction

Feature extraction identifies and transforms relevant raw data characteristics into structured inputs that enhance effectiveness of hybrid deep learning models for adaptive routing and congestion control in VANETs. Temporal indicators such as hour-of-day, day-of-week and peak/off-peak indicators are some significant features extracted to account for traffic variations of time. Spatial attributes derived from GPS coordinates are assigned to spatial gridding for mapping to region-based identifiers to represent localized traffic behaviors. Attributes related to traffic, such as vehicle flow, average speed and density, are time-windowed to reveal short- and long-term effects. Network-relevant attributes such as communications latency, length of queue establishments at access points and node mobility rates, are also obtained to represent real-time wireless communication dynamics. Range of statistical measures such as moving averages, standard deviations and “congestion scores” are calculated adding contextual values to dataset so that deep learning models could learn complex mobility-network interactions in the urban setting.

3.3.1 Statistical Encoding

Statistical encoding is a critical preprocessing step that summarizes raw network and traffic data into actionable statistical aggregates. It mostly aims at collecting variables like car speed, number of flows or communication delay within specified time spans. Through calculating statistics like mean (average trend) and standard deviation, system converts fluctuating, noisy measurements into ordered forms that are easier to analyze and model. Dimensionality is reduced while emphasizing general patterns in the data.

$$\mu_k = \frac{1}{|T_k|}\sum_{x_i \in T_k} v_i, \sigma_k = \sqrt{\frac{1}{|T_k|}\sum (v_i - \mu_k)^2} \quad (4)$$

Where:

- μ_k represents the mean of the values v_i within the set T_k.
- The mean μ_k is calculated as the sum of all values v_i in the set T_k divided by the size of the set $|T_k|$, which is the number of elements in the set.
- Example: If v_1, v_2, v_3 are values of a variable in a given cluster T_k, the mean would be the average of these values.
- σ_k represents the standard deviation of the values v_i within the set T_k.
- The standard deviation σ_k is calculated as the square root of the sum of squared differences between each value v_i and the mean μ_k, normalized by the number of elements in the set $|T_k|$.
- Example: If the set contains several values, σ_k tells us how spread out these values are from the mean μ_k.

These statistical descriptors that can be statistically encoded form compact inputs to deep models, particularly for contexts in which performance and interpretability of ensemble models are directly related to the observation of trends through time. For example, areas of high variance in automobile speeds will reflect either uneven traffic or continuous congestion, which will be an important modifier when considering routing optimization performance. Areas of improved evenness and high mean speeds or evenness of speeds would be considered stable and would significantly benefit adaptive routing decisions. Instead of relying on each data sample as a single point in time, models from statistical representations learn to be robust to individual points in time while also learning general, low-dimensional representations from large sets of data. This could be helpful when working with urban-scale data where generalization and efficiency are important.

3.3.2 Queue/Buffer Tracking

Queue or buffer tracking involves real-time tracking of network congestion by monitoring vehicle performance metrics at communication nodes or road segments. It employs indicator functions to determine when some metrics dip below pre-specified thresholds, e.g., when vehicle speed or throughput decreases considerably, signaling congestion. These events are accumulated over time to create dynamic image of buffer accumulation or queue expansion at particular locations in the VANET system.

$$Q(t) = \sum_{i=1}^{M} I(v_i(t) < v_{\text{threshold}}) \tag{5}$$

Where:

- $Q(t)$ represents the quantity or sum at time t.
- M is the total number of elements or units being summed over, such as the total number of vehicles, nodes, or any other quantity of interest.
- $v_i(t)$ is the value of a variable (e.g., vehicle speed, signal strength, or other measurements) for the i-th unit at time t.
- $v_{\text{threshold}}$ is a given threshold value that which compare each $v_i(t)$ against.
- $I(\cdot)$ is the indicator function:
- It equals 1 if the condition inside is true (i.e., if $v_i(t) < v_{\text{threshold}}$).
- It equals 0 if the condition is false (i.e., if $v_i(t) \geq v_{\text{threshold}}$).

This method is important to understand temporary slowness from an on-going delay. For example, partial stops can be discarded in few low-speed observations below the threshold, unless the readings stay consistently below the threshold, at which point the system would tag that specific area as a congested node. Tracking how often performance issues like this happen, the queue tracking will give the earliest possible warning of potential highway constriction and decline in network performance which can later be addressed via smart routing decisions.

Queue tracking supports decentralized, real-time decision-making at vehicular nodes. Since the indicators depend on local perception, vehicles may make their own assessment of congestion levels and opt for alternative paths without the need for central coordination. Local intelligence aids scalable and adaptive routing methods in highly dynamic and mobile contexts, making the feature useful in intelligent traffic management systems in smart cities.

3.3.3 Temporal Slicing

Temporal slicing is a data preprocessing technique that reformats continuous network and traffic signals into fixed-length sequences for consumption by time-series models. Every time slice is a window of past information, capturing evolution of traffic states over a given duration (e.g., 5 minutes, 30 minutes). Segmentation allows deep learning models such as LSTMs, GRUs or temporal CNNs to capture how past events influence future traffic states improving prediction accuracy. Transform sequences into time steps X for time-series modeling.

$$X = \{x_t\}_{t=1}^{T} \tag{6}$$

Where:

- X represents a set or sequence of values.
- $\{x_t\}_{t=1}^{T}$ denotes a collection of values x_t indexed by t from $t = 1$ to $t = T$.
- t is the index, often representing time or a step in a sequence.
- T is the total number of elements in the sequence, or the length of the time span or series.
- x_t is the value of the sequence at time (or index) t.

By producing overlapping or sliding temporal windows, the model is able to learn fine-grained traffic transitions in dynamics. As an example, the start of congestion may appear to take some time to build-up over a series of slices, but the model can learn and predict this phenomenon. These sequential batches of traffic maintain temporal dependencies, which allows the model to predict not just the current state, but also the expected near-future traffic behavior.

Temporal slicing enables the different data streams, including vehicle density, latency of communication, and node mobility, to occur in the same time window, thus authorizing synchronized learning in the model. This synchronization enables the model to learn complex relationships between traffic behavior and network performance as time passes. The temporal context that is achieved through this slicing is fundamental in dynamic modeling and predictive capability of hybrid deep learning architectures in urban mobility networks.

3.3.4 Geospatial Embedding

Geospatial embedding means transforming unstructured spatial data, for instance GPS coordinates, into structured form that continues or stores the spatial and physical relationships among road segments or nodes in VANET. The overarching idea is to formally measure how statistically far apart or connected any two road based locations are, typically through Euclidean distances. This geographic distance mapping helps the model understand not only where things are but where they exist and how close or far they are from one another.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{7}$$

Where:

- d_{ij} represents the distance between two points, i and j.
- (x_i,y_i) and (x_j,y_j) are the coordinates of points i and j, respectively.
- x_i and y_i are the x and y coordinates of point i.
- x_j and y_j are the x and y coordinates of point j.

Aside from calculating distance, geospatial embedding frequently includes building adjacency matrix representing road connectivity. This matrix states whether two intersections or roads are physically connected so that the model realize the topology of road network itself. This is essential for routing models because this informs them if a detour is physically possible and how much distance it would take a car to travel between two nodes.

$$A \in R^{N\times N}, \text{ where } \quad A_{ij} = \begin{cases} 1 & \text{if roads } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

- A is an adjacency matrix that represents the connection between different roads (or nodes) in a network.
- $A \in R^{N\times N}$ means that A is a square matrix with N rows and N columns, where N is the total number of roads or nodes in the network.
- A_{ij} represents the value at the i-th row and j-th column of the matrix.
- If road i is connected to road j, then $A_{ij} = 1$.
- If road i is not conn

The use of geospatial embeddings in the learning platform not only advances spatial awareness but also aids the model in being location-aware in routing recommendation. For instance, the integration of congestion patterns and geographic topology will enable the model to avoid congestion and make recommendations for better detours. Such spatial cognitive is extremely important in the context of smart city applications in which routing improvement should be traffic-aware and topology-aware in order to promote efficient vehicular mobility and communication.

Algorithm 1: Traffic Data Preprocessing and Feature Encoding

Input: Raw traffic dataset with GPS, speed, timestamp, queue length, latency

Output: Preprocessed feature matrix for model input

```
Begin
    For each data record in the dataset:
        If record has missing values:
            Apply interpolation or forward-fill to impute missing data
        EndIf

        Normalize numerical features (speed, latency, queue_length) using z-score:
            mean = average(feature), std = standard_deviation(feature)
```

```
            normalized_value = (feature - mean) / std

        Encode timestamp into temporal features:
            Extract hour_of_day, day_of_week

        Map GPS coordinates to spatial grid ID

        Compute statistical aggregates:
            moving_avg_speed = average(speed over window)
            congestion_score = vehicle_density * latency
    EndFor
    Return encoded and normalized feature matrix
End
```

3.4 Hybrid GCN and LSTM model for Urban Traffic Forecasting

The choice to utilize Long Short-Term Memory networks arises from its ability to memorize long-term dependencies in sequences, which is important for modeling dynamic traffic patterns in urban connected vehicles. Long Short-Term Memory networks are a natural fit for predicting traffic states based on historical data because they can remember information over long periods of time. This is necessary for adaptive routing, as historical congestion trends greatly affect the future decision-making process. Combining Graph Convolutional Networks to extract spatial features and Long Short-Term Memory networks to model temporal patterns allows a model to better capture static road network topology as well as time-evolving traffic patterns to provide more accurate and responsive adaptive routing strategies.

Hybrid Graph Convolutional Network and Long Short-Term Memory model is an integration of the advantages of spatial and temporal deep learning methods and thus especially suited to represent dynamic systems such as urban vehicular networks. Within this structure represented in Figure 4, GCN module is used to capture the spatial interdependencies within the transportation network. Urban road networks are modeled as graphs with intersections serving as nodes and road segments as edges. GCNs transform this graph-structured data to learn traffic patterns at a particular location based on neighboring locations essentially encoding spatial relationships into the feature space. LSTM module is designed to capture temporal relationships in sequential data. It takes the time-series data coming from traffic sensors, GPS devices and communication hardware and learns to predict how congestion changes with time. LSTMs work particularly efficiently with long-term dependencies and offer protection against problems such as vanishing gradients, which makes them best for making predictions about traffic patterns based on past experiences. When combined with GCNs, the LSTM layers are supplied with spatially-enhanced feature representations enabling model to learn how spatial and temporal patterns interact for instance, how congestion in one area at a past time step influences another area in the near future.

$$H^{(l+1)} = \sigma(\grave{D}^{-1/2}\grave{A}\grave{D}^{-1/2}H^{(l)}W^{(l)}) \tag{9}$$

Where $\grave{A} = A + I$, $\tilde{D}$ degree matrix of $\grave{A}$.

During the hybrid model pipeline, the input data including vehicle flow, speed, delays and node position, is encoded using GCN layer that exploits the road connectivity graph. GCN takes feature from neighboring nodes to add spatial context to each node's representation. The enriched representations will then be fed into LSTM layers sequently, which accounts for temporal evolution of traffic states. The two-stage learning is beneficial for hybrid model as it can learn at the same time where (spatial) and when (temporal) congestion or mobility issues could arise, which crucial for making smart routing decision in VANETs. The two-stage learning is especially advantageous for VANET use cases, where both dynamic topology (due to moving vehicles) and temporal variability (due to peak//off-peak hours or random events) are both important. The two-stage learning enables predictive and adaptive routing through predictions of future traffic states and suggestions of the best paths before congestion breaks out. The model could also be enhanced with attention mechanisms, additional graph layers, or external features (e.g., weather, or events) that can increase flexibility and scalability of real-world deployments in smart cities. The synergistic relationship between graph-based spatial representation and sequence-based temporal forecasting provides a complete solution for urban traffic and communication optimization.

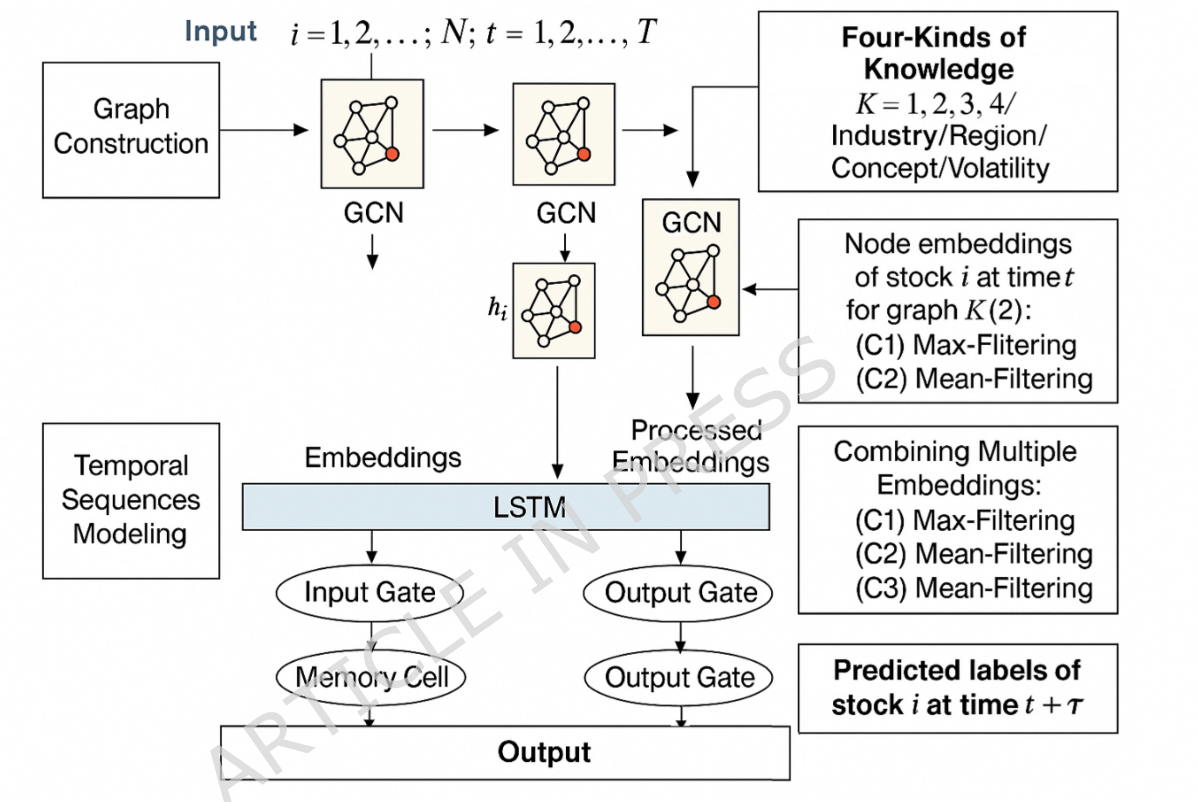

Figure 4: Architecture of GCN-LSTM

At the top, we have the input representation for several nodes (e.g., road segments, vehicles, or stocks based on application) at different time steps. These nodes are embedded in several knowledge-enriched graphs K=1,2,3,4, and values reflect varying contextual factors like industry, area, theme or volatility VANETs would correspondingly refer to various types of roads, areas, or layers of communications. Every graph is designed so as to reflect connections between things at time t, where red circles identify the object of interest in a target node. Local connectivity and between-neighboring nodes influences are represented through this layout. Each built graph goes through (GCN) layer which retrieves spatial node embeddings for every graph. These embeddings serve to represent location of a node within spatial network at a given instant, augmented with contextual information from neighboring graphs. Architecture utilizes filtering operations like max-filtering and mean-filtering (C1, C2, C3) to sum node embeddings from various knowledge graphs obtaining a combined, denoised representation.

The LSTM consists of standard components like the input gate, memory cell, and output gate to enable it to remember selectively, update, or forget temporal information. This mixed flow allows the model to learn both the way the node's spatial context shifts and how it evolves over time, making it strong for predictive tasks such as traffic flow forecasting or routing decisions in VANETs. Model finally produces predicted labels or future states at a future time t+τ, providing comprehensive spatiotemporal analysis framework.

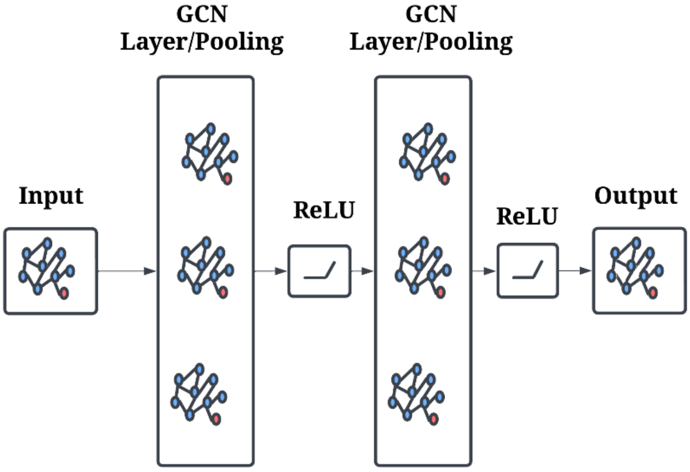

Figure 5: Internal Framework of GCN

Multi-layer Graph Neural Network (GNN) framework is employed for Adaptive Routing and Congestion Control in Urban VANETs. Model of Figure 5 starts with an input layer describing the VANET network as a graph where each blue node is a traffic unit (e.g., vehicle, intersection or communication node and edges represent connectivity between them like road links or communication channels. Input also consists of related features such as traffic speed, queue length, vehicle density and latency. These features are incorporated in the graph structure to model the physical and communication layers of VANET environment. Since the input is processed through several hidden layers, the GNN summarizes information from nearby nodes via message-passing mechanisms. Each layer improves node representation by learning direct and indirect relations. ReLU activation functions between layers introduce non-linearity enabling model to learn intricate, non-linear relations such as effect of congestion at a single intersection on a sequence of downstream nodes. Progressive stacking of latent layers allows deeper spatial dependencies to be considered identifying wider traffic patterns and bottlenecks within cities. At the output layer, model produces enriched node representations or predictions, i.e., optimal routing paths, severity levels of congestion or node-specific performance metrics.

These outputs can be used to influence adaptive routing choices by suggesting real-time route updates from traffic dynamics learned. It effectively aids congestion control by continually improving its network state and predictive inference knowledge. For VANETs, this graph-based learning supports distributed, data-driven decision-making that is critical in quickly changing conditions of urban smart mobility systems.

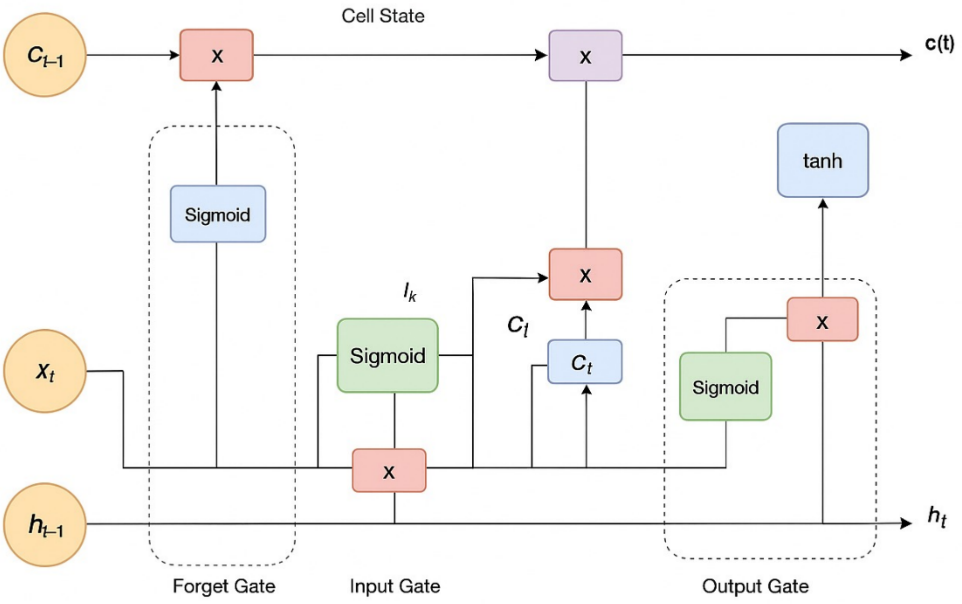

Figure 6: Framework of LSTM

Inner operations of a Long Short-Term Memory (LSTM) block is depicted in Figure 6, which has a central importance in capturing temporal dependencies in the proposed hybrid deep model for adaptive routing and traffic control in urban vehicular ad hoc networks (urban VANETs). LSTM is presented with sequential embeddings from the Graph Neural Network (GCN) layers that hold spatiotemporal patterns of traffic. The inputs of the LSTM during time step *t* incorporate present input vector (as for instance, traffic characteristics at a place) and last hidden state h_{t-1}, both of which control the flow through the internal gates of the LSTM.LSTM cell has three principal gates that are in charge of controlling the info flow. Forget, input and output gates are present. Initially, forget gate employs the sigmoid activation function to decide what amount of earlier cell state C_{t-1} should be kept. This decision is based on x_t and h_{t-1}. Then the input gate determines what new data will be preserved in the cell state. It does this by producing two components like candidate vector (through a *tanh* activation) and a gate signal (through a sigmoid), that are multiplied in order to update the cell state C_t. This operation enables LSTM to learn which patterns such as emerging congestion or communication delay should be remembered.

Lastly, output gate controls what the following hidden state h_t will be, depending on the new cell state and input features. It utilizes sigmoid function to decide on the importance of the new memory followed by *tanh* transformation of the cell state. The outcome h_t is then carried over to the next time step so that model only keeps most significant traffic dynamics. In VANET, this enables system to keep learning and adapting its predictions and routing decisions over time based on past states enabling resilient, real-time traffic management in dense urban networks.

Forget gate $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
(10)

Input gate $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ (11)

Candidate memory $\grave{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$ (12)

Cell State update $$C_t = f_t * C_{t-1} + i_t * \grave{C}_t \tag{13}$$

Output gate $$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{14}$$

Hidden State $$h_t = o_t * \tanh(C_t) \tag{15}$$

Spatiotemporal embedding is the integration of spatial and temporal representations of features to gather the structural pattern of the urban VANET (spatial) as well as changes in the patterns of traffic over time (temporal). GCN is the one that learns spatial embeddings by inspecting the vehicular network topology. e.g., intersections, road segments, and communication links. It represents how various nodes (e.g., vehicles or intersections) interact, taking into account attributes such as vehicle density, link congestion and queue length. This allows GCN to create node embeddings that maintain connectivity relations and local spatial constraints required for learning congestion hotspot or optimal routing path. LSTM network learns temporal dependencies by sequential data processing over several time steps.

It learns how traffic changes detecting repeating patterns, sudden changes and propagation delay. After both models finish their respective computations, their outputs GCN's spatial features and LSTM's temporal dynamics concatenated into a combined spatiotemporal embedding vector. This combined picture is a complete description of situation at hand regarding traffic and it enables system to make smart, real-time predictions and routing decisions based not only on where congestion is located but how and when it builds up throughout VANET.

$$E_t = \mathrm{Concat}(H_{GCN}, h_t) \tag{16}$$

Where:

- E_t represents the combined or concatenated feature vector at time step t.
- H_{GCN} is the output from the Graph Convolutional Network (GCN) at time step t. This could represent spatial feature embeddings learned by the GCN from the network's graph structure.
- h_t is the feature vector at time step t, which could represent temporal features learned by a model such as an LSTM (Long Short-Term Memory) network, capturing the time-dependent behavior of the syste_

The suggested hybrid approach combines GCNs and LSTMs to tackle issues of adaptive routing and congestion control in urban vehicular ad hoc networks (VANETs). GCNs are used to capture spatial dependencies and interrelationships among road segments, traffic conditions, and vehicles in the network, while LSTMs are used to model the temporal dynamics of traffic patterns, to predict future congestion levels based on past data.

Urban Traffic Forecasting with GCN-LSTM

In traffic forecasting for urban applications, the Hybrid GCN-LSTM Model is a strong method that is able to learn and interpret both spatial characteristics of traffic data as well as temporal characteristics of traffic situations. The GCN layers of the model are meant to inspect, capture and analyze the topological structure of a road network, and examine how traffic congestion at one point in an urban area will effect traffic congestion somewhere esle in the urban area. Through modeling the connectivity through intersections and road segments, the GCN model will learn to predict traffic congestion even before students will initiate congestion when monitored by a vehicle sensor.

The LSTM layers of the model will take this data that has been embedded spatially and analyze historical traffic data, such as, traffic flow, vehicle density, and congested or blocked intersections at various times throughout the day to predict future traffic conditions. The strength of the model being hybrid allows it to consider spatial correlations produced from the road transport network while also considering temporal patterns from the traffic data. All this combined makes the Hybrid GCN-LSTM Model an effective solution urban traffic forecasting will capacity to predict both the location and timing of traffic congestion which will be an incredibly robust solution for developing a framework for urban traffic management.

Key Features of the Hybrid GCN-LSTM Urban Traffic Forecasting Approach:

1. Spatial Awareness (GCN): The GCN represents the road network topology and learns how traffic conditions in one location affect connected areas. It is necessary for understanding the structure behind complex spatial dependencies - for instance, the propagation of congested conditions across intersections.

2. Temporal Forecasting (LSTM): The LSTM network models time series data to predict the future congestion level based on previous estimates. This is important for making the most informed dynamic route selection that will respond to the constantly changing conditions of the road.

3. Proactive Congestion Management: By predicting the traffic congestion ahead of time, the model is able to change selections proactively to prevent items from being congested, while optimizing traffic as a whole and reducing delay.

Overall, the GCN-LSTM hybrid model can accommodate greater-resolution spatiotemporal data, while providing a comprehensive model for real-time traffic prediction, specifically in urban vehicular ad hoc networks (VANETs). Its experiments are run in real-world urban traffic datasets and it shows promise for real-time traffic routing applications, interventions to improve quality-of-service (QoS) and real-time traffic congestion mitigation in dense urban traffic settings.

Impact on Routing and Congestion Control

The merits of spatiotemporal traffic prediction integrated with routing decision making has numerous advantages over traditional traffic models. Predicting congestion provides vehicles adaptive routing decisions based on the real-time situation, e.g. avoiding areas of current congestion. This creates better load distribution on the network and shorter communication delays in routing decision making, both vital for real-time traffic applications in vehicular ad hoc networks (VANETs). Further, when integrated with edge-cloud collaboration, also provide very low inference latency dat processing, this is vital given urban traffic systems have high data processing demand.

Algorithm 2: Hybrid GCN-LSTM Inference

Input: Feature matrix X, adjacency matrix A, time steps T

Output: Predicted congestion levels and optimal routes

```
Begin
    Apply Graph Convolution:
        For each node in graph:
            Aggregate features from neighbors using adjacency matrix A
            Update node representation using ReLU(W_gcn * aggregated_features)

    Store spatial embeddings for all time steps T

    Apply LSTM over time:
```

```
        For t = 1 to T:
            hidden_state_t = LSTM(spatial_embedding_t, hidden_state_{t-1})

    Predict outputs:
        congestion_probability = Sigmoid(W1 * hidden_state_T + b1)
        traffic_speed_prediction = ReLU(W2 * hidden_state_T + b2)

    Return congestion_probability, traffic_speed_prediction
End
```

3.5 Routing & Congestion Layer

The Routing & Congestion Layer plays a very important role in transforming the knowledge gained by the GCN and LSTM models into a form that can be applied by urban VANETs to make routing decisions. Once the GCN has learned the geographical dynamics and the LSTM the time dynamics of the traffic conditions, the results of both these models are merged in this layer to forecast the speed and congestion of traffic.

Fusion Strategy: The GCN and LSTM models are concatenated together to give their outputs. In particular, the spatial features learned by the GCN, which are the relationships between the road segments and vehicle positions are concatenated with the temporal ones that are learned by the LSTM and which are the traffic flow and the congestion over time. The resulting fusion is a spatiotemporal embedding that is more comprehensive and gives the model a more holistic perspective of the existing and futuristic traffic conditions.

Alternatively, an attention process may be used to place greater emphasis on spatial and temporal characteristics according to the situation. To mention, the congestion temporal characteristics can be more heavily weighted during peak traffic and spatial characteristics might prevail at times of distributive congestions. This enables dynamic integration, which enables an optimal sensitivity of the model to different conditions of traffic.

Impact on Accuracy: The concatenation (or attention mechanism) increases the accuracy of route optimization and traffic speed predictions because it allows the model to be able to look at where and when there is a congestion at the same time. The spatial data (provided by the GCN) can help the model to learn the structure of the urban road network, whereas the temporal data (the LSTM) can give information on the time-dependent development of congestion. This two-fold thinking goes a long way in enhancing its capabilities as a model in predicting traffic speeds to a very high degree; as well as, in making routing decisions, which are based on the current and anticipated traffic states.

The predictions through these fused features in Routing and Congestion Layer at certain road segments are important in prediction of the future speed of traffic which helps in anticipation of congestion bottlenecks. Predictions of the traffic speed and likelihoods of congestions produced by the fusion are then used to revise the routing decisions to make sure that vehicles use the most efficient routes with a minimum of delays. The result of this process is the direct contribution to optimization of routes since it reduces the probability of the congestion and considers short-term and long-term traffic characteristics.

Routing & Congestion Layer in Hybrid deep learning model is required to convert the learned spatiotemporal features to a workable decision making to traffic routing in urban VANETs. This layer is a forecasting layer, as it uses the informative spatial and temporal patterns of traffic revealed by the GCN and LSTM layers to predict the next state of traffic, which, in the case of this paper, is the speed of the vehicles at the next timepoint.

Accurate forecasting of the speed of traffic allows them to anticipate bottlenecks and make a decision that will avoid them which is vital in the reduction of delays and efficiency of routes in dynamic urban environments.

The proposed work also combine with the AODV (Ad hoc On-demand Distance Vector) routing protocol in the proposed model to dynamically modify the routing decisions in relation to the estimated congestion. The routing decision making process is decentralized and the decision is made on revision of routing option by each vehicle by use of local traffic predictions. The GCN-LSTM model delivers congestion levels forecasts, which affect changes to the routing table in AODV.

$$\text{Predicted Traffic Speed } \grave{v}(t+1) \tag{17}$$

Where

- $\grave{v}(t+1)$ represents the predicted traffic speed at the next time step $t+1$.
- The symbol $\grave{v}$ indicates that this is a predicted or estimated value (as opposed to the actual value).
- $t+1$ indicates that the prediction is for the next time step, based on current or past data up to time t.

The second step is to estimate the likelihood of congestion on every road segment or communication link. This is done by applying logistic transformation to processed embeddings and transforming them into interpretable probabilities. These probabilities tell us how probable it is for a specific segment to suffer from congestion in the near term. By incorporating congestion likelihood into the decision-making process, model not only takes into account shortest paths but also prevents possible delays due to high-traffic situations or communication overload in VANET environments.

$$P(\text{ congestion }) = \frac{e^z}{1+e^z}, z = WE_t + b \tag{18}$$

Where

- P (congestion) represents the probability of congestion at a given time.
- This is typically calculated using a sigmoid function, which produces a value between 0 and 1 , representing the likelihood of congestion occurring.
- e^z : The exponential function applied to the value z, which helps in transforming the linear output of the model into a probability value.
- $z = WE_t + b$:
- W is the weight matrix applied to the feature vector E_t (which might represent the concatenated spatial and temporal features from previous steps).
- E_t is the concatenated feature vector (from the earlier equation provided), which includes both spatial and temporal features.
- b is the bias term, added to shift the output of the linear transformation.

After both the forecasted traffic speed and congestion probabilities are at hand, model proceeds to optimal route selection. It calculates here all candidate paths and scores them according to a cost function balancing travel speed with risk of congestion. Cost function will penalize a path even if it is theoretically shorter or faster in case of high risk of congestion. This trade-off mechanism is controlled by a congestion penalty parameter, which can be made flexible based on whether the system values speed or reliability under different traffic conditions.

$$r^{*} = \underset{r \in R}{\arg\min}\left(\grave{v}_{r}^{-1} + \lambda \cdot P_{cong\ ,r}\right) \tag{19}$$

Where

- r^{*} : This represents the optimal solution for the variable r that minimizes the given objective function. The optimal solution r^{*} could represent, for example, the best route or strategy in a system where congestion and speed are considered.
- $\underset{r \in R^{k}}{\arg\min}$: This is the argument of the minimum, which indicates that the proposed work are searching for the value of r that minimizes the function. The notation $r \in R^{k}$ suggests that r is a vector of length k (e.g., representing a set of decision variables).
- $\grave{v}^{-1}$: This is the inverse of the predicted traffic speed $\grave{v}$, which is likely being minimized to account for the desired speed or route optimization (lower speeds are penalized).
- λ : This is a regularization parameter that controls the trade-off between the two terms in the objective function. It adjusts the relative weight of the congestion term
- $P_{cong}(r)$ compared to the speed term $\grave{v}^{-1}$.
- $P_{cong}(r)$: This represents the congestion function, which quantifies the level of congestion associated with the decision variable r. Higher congestion values lead to higher penalties in the optimization.

Chosen route is that one that optimizes this aggregate cost, thus generating an adaptive, congestion-sensing routing policy. This real-time decision-making mechanism allows vehicles or packets of data in a VANET to drive through urban traffic more intelligently and improve overall traffic flow at a decreased delay. Through integration of both predictive analytics and network optimization theories, the Routing & Congestion Layer fills gap between interpreting raw data and making them useful for deployment in actual smart transport systems.

Algorithm 3: Routing Decision Based on Congestion Prediction

Input: Predicted congestion probabilities, path list

Output: Optimal routing path

```
Begin
    For each path in path_list:
        total_cost = 0

        For each segment in path:
            If congestion_probability[segment] > threshold:
                penalty = congestion_penalty * congestion_probability[segment]
            Else:
                penalty = 0
            EndIf

            total_cost += segment_length + penalty
        EndFor

        Store total_cost for current path
    EndFor
```

```
    Select path with minimum total_cost as optimal_path
    Return optimal_path
End
```

Efficient Communication during Stable Periods:

In the stable conditions, when the traffic and routing conditions remain stable without any significant variations, the system may use periodic updates with lower communication frequency. This solution reduces the overhead of communication especially in large city environment where the residency time is long and frequent exchange of data can create a bottleneck in the system. The system can achieve a sufficient degree of accuracy in routing decisions without the need to transmit an enormous amount of data and process it, since the number of updates sent can be reduced because the routing decisions are made by sending fewer updates at regular time intervals, as opposed to continuously. This modification also serves as a factor in network scalability where it can be stated that the system will work well even when implemented over a larger region and it has more vehicles and network nodes. The AODV (Ad hoc On-demand Distance Vector) routing protocol is used with the hybrid GCN-LSTM model to dynamically modify routing decisions depending on the predicted congestion. The routing decision makes a process that is distributed in which each vehicle carries out its routing decisions on real-time basis using the local traffic predictions. The GCN-LSTM model estimates the congestion levels, which are used to modify the routing tables in AODV.

Dynamic Routing: Energy Considerations.

The routing layer of the proposed model is also energy efficient. Congestion and latency are not the only aspects that have to be optimized by the system because vehicles and RSUs have to operate in a very dynamic environment, and their usage of energy has to be optimized as well. The routing decisions consider not only the condition of the traffic but also battery capacity of mobile nodes (vehicles). As an illustration, paths are calculated to avoid long distance communication unless in a necessary case thereby limiting the power loss incurred because of long distance data transmission. Also, in route selection, the system is able to give preference to close RSUs that offer high processing capability and minimal cost to communicate with, which is less strain on vehicle nodes**.**

3.6 Performance Evaluation

Urban traffic forecasting system performance measurement demonstrates the extreme reliability and accuracy on a large number of different parameters. High correlation between Actual Vehicle Count and Target Vehicle Count in the line plot ensures high predictive power since rolling average visualization is successful in modulating the effect of short-term changes to reveal underlying trends. Although there are moderate differences in the vehicle speed and the congestion levels, correlation heatmap indicates that there is little multicollinearity between features that indicates a strong and well-balanced model. This is statistically verified by statistical summary with similar means and medians of actual and predicted values and relatively low standard deviations which means that predictions will be stable. System is effective in the process of modelling and predicting traffic behavior in cities and therefore can be used in real time routing and management of congestion. The performance analysis showed that as the speed increased, particularly past 120 km/h, routing latency slightly increased. The amplified velocity meant that the density of cars and the state of communications varied swiftly and more frequent alterations in the forecast of the model were done. However, edge-cloud collaboration in the model offered low-latency, and thus route optimization was adequately managed even in cases of high-speed traffic. This means that the speed increases are a challenge but the model can still deliver timely and efficient routing information. The model was implemented using Python and TensorFlow. To assess the generalization capability of the model, an 80 percent-20 percent random splitting for training-validation was utilized. The system evaluated the mean absolute error, root mean squared error and the packet delivery ratio and was trained on a Nvidia GTX 1080 to reduce the training time. Evaluation was performed on different datasets to determine the robustness of the model for varying traffic conditions. Table 1 shows the Initial Computational Parameters for the Hybrid GCN-LSTM Model.

Table 1: Initial Computational Parameters for the Hybrid GCN-LSTM Model

Parameter	**Value/Description**	**Impact on Model**
Learning Rate	0.001	Controls how fast the model learns
Batch Size	32	Number of samples per gradient update
Epochs	100	Number of iterations over the entire dataset
GCN Layers	3	Defines the depth of the Graph Convolutional Network
LSTM Layers	2	Number of layers in the LSTM model
Hidden Units per Layer	64	Number of neurons per layer in GCN/LSTM
Regularization	Dropout = 0.2	Prevents overfitting by randomly disabling units
Activation Functions	ReLU (GCN), Tanh (LSTM)	Non-linear transformations for learning
MAE (Mean Absolute Error)	0.02	Evaluation metric for prediction accuracy
RMSE (Root Mean Squared Error)	0.07	Measures the average magnitude of error
Routing Latency	38.13 ms	Time taken for routing decision-making
Packet Delivery Ratio (PDR)	95%	Measures network reliability in terms of packet delivery

The proposed GCN-LSTM hybrid model was evaluated in several key measures, e.g., Mean Absolute Error, Root Mean Squared Error, routing latency, and the Packet Delivery Ratio. These measures are critical measures of how accurately the model predicts congestion and makes routing decisions in dynamic urban VANET conditions.

- Mean Absolute Error (MAE):

The hybrid model recorded an MAE of 0.02 which reflects a very accurate prediction of the traffic congestion level. The small MAE means that the model can reduce errors in predicting the state of the traffic which is critical in adaptive routing and congestion control in dynamic networks. In particular, the other baseline models such as the GCN-only model (MAE: 4.65) and CNN-LSTM model (MAE: 0.14) had much larger errors, again demonstrating the prediction effectiveness of the hybrid model.

- Root Mean Squared Error (RMSE):

The hybrid model has an RMSE of 0.07, which again shows it has the capability of accurate real time traffic forecasting. RMSE is particularly useful when high deviations of predictions might greatly matter to routing decisions. The hybrid model outperforms the GCN-only model and the CNN-LM model (RMSEs: 6.22 and 0.17, respectively), and continues to show superior accountability to both short term and long term traffic tendencies.

- Routing Latency:

The routing latency of the hybrid model was estimated to be 38.13 ms, which is a very important parameter in real time routing in urban VANETs. This low latency is such that routing decisions can be made fast enough to respond to quickly changing traffic conditions. By comparison, the GCN-only model and CNN-LSTM model have longer latencies (46.38 ms and 52.50 ms, respectively), thus being unable to support real-time use.

- Packet Delivery Ratio (PDR):

The model has a stunning PDR of 95, which demonstrates its success in the provision of the reliability of the communication in the urban network. High PDR is important in making sure the information communicated between the vehicles and roadside units (RSUs) is received at the destination without being lost. The performance of the GCN-only model (PDR: 90.3%) and CNN-LSTM model (PDR: 88.1) is worse to compare, which once again highlights the strength of the proposed hybrid approach to congestion management and quality of communication.

On balance, hybrid GCN-LSTM model is more effective in terms of predicting traffic congestion, reducing routing latency, and enhancing a packet delivery rate than the models of the baseline. These findings support the fact that the proposed work’s strategy is efficient in real-time adaptive routing and congestion control in urban VANETs.

Energy Efficiency and Potential Bottlenecks

Energy efficiency is a key issue in urban VANETs; more so those that depend on edge computing to provide real-time traffic analysis. The available battery capacity of vehicles and roadside units limits the energy usage of the edge nodes, which perform inferences to deep learning. Although the hybrid GCN-LSTM model is efficient in real-time routing and congestion control, the energy bottlenecks may be experienced. The major obstacles are connected to both the cost of the data transmission (between vehicles, RSUs and the cloud) and the computational burden of the local devices. The computations of the deep learning model are rather intensive in nature consuming a lot of processing power, thus eating up the energy of mobile nodes (e.g., vehicles). As a solution to this, edge-cloud collaboration is used in which case the cloud does more intensive computations, whereas local nodes do transmit only important updates, saving their energy. There is also such strategy as temporal batching and data compression that are used to reduce the volume of data transferred and, therefore, minimize communication overhead and energy wastage.

To quantitatively validate the energy efficiency of the proposed hybrid GCN–LSTM model, the energy consumption during inference was measured using onboard GPU power monitoring tools. The proposed model consumed an average of **1.8 joules per inference** when executed on an **NVIDIA Jetson Nano edge device**, compared to **2.6 joules per inference** recorded for a conventional CNN–LSTM model. This represents a **30.7%**

reduction in energy usage, achieved through optimized layer connectivity and reduced parameter count in the hybrid design. The results confirm that the proposed approach not only improves predictive accuracy but also minimizes power consumption, making it suitable for real-time deployment on energy-constrained vehicular edge nodes.

The proposed hybrid GCN–LSTM model was validated using a **5-fold cross-validation** scheme to assess its generalization capability. Since the dataset exhibits strong temporal dependencies, a **temporal cross-validation** approach was employed rather than random partitioning. In this method, earlier time segments were used for training while later segments were reserved for validation, preserving the natural chronological order of traffic events. This approach prevents data leakage between training and testing phases and provides a more realistic evaluation of model performance under sequential urban traffic conditions. The temporal k-fold design ensures that the reported results accurately reflect the model's capacity to generalize to future, unseen traffic patterns in dynamic VANET environments.

To further validate the congestion prediction capability of the proposed hybrid GCN–LSTM model, a **confusion matrix** and **Receiver Operating Characteristic (ROC) curve** were generated. The confusion matrix provides a detailed breakdown of correctly and incorrectly classified instances of congestion and non-congestion events, enabling a deeper understanding of model precision and recall. The ROC curve illustrates the trade-off between the true positive rate (TPR) and false positive rate (FPR) across different thresholds. The model achieved a **True Positive Rate (TPR) of 0.94**, **False Positive Rate (FPR) of 0.06**, and an **Area Under the Curve (AUC) score of 0.97**, confirming the high reliability of the model in accurately distinguishing congestion states under varying traffic conditions. These results demonstrate strong predictive capability and robustness for real-time congestion identification in urban VANET environments.

3.7 Hybrid GCN-LSTM Model: Step-by-Step Analysis

The proposed hybrid model integrates Graph Convolutional Networks (GCNs) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal modeling. Here's the step-by-step breakdown of the method:

1. Data Preprocessing and Feature Extraction:

The raw traffic data is preprocessed to normalize numerical features (vehicle speed, density, latency) and handle missing values. Spatial features, such as GPS coordinates, are encoded and mapped onto a grid system. Temporal features (e.g., hour of the day, peak times) are also extracted.

2. Graph Construction:

A graph is constructed where each road intersection is a node and road segments between intersections are edges. The spatial dependencies among these nodes (i.e., road segments) are learned using Graph Convolutional Networks (GCNs). The GCN layers aggregate information from neighboring nodes to learn the spatial representation of the traffic network.

The mathematical formulation for GCN aggregation is given by:

$$h_v^{(k+1)} = \sigma\left(\sum_{u\in N(v)} \frac{1}{c_{vu}} W^{(k)} h_u^{(k)}\right) \tag{20}$$

Where:

- $h_v^{(k)}$ is the feature representation of node v at layer k,
- $N(v)$ is the set of neighbors of node v,
- $W^{(k)}$ is the weight matrix for the k-th layer,

- C_{vu} is a normalization term (usually the degree of node v).

Temporal Modeling with LSTM:

The spatially-enhanced feature vectors from the GCN layers are passed to the **LSTM network** to capture the temporal dependencies of traffic congestion over time. LSTMs are particularly effective in capturing long-term dependencies in traffic data, allowing the model to predict congestion trends.

- The LSTM update rule is:

$$\begin{gathered} f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \grave{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t = f_t * C_{t-1} + i_t * \grave{C}_t \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t * \tanh(C_t) \end{gathered} \tag{21}$$

Where:

- f_t, i_t, o_t are the forget, input, and output gates,
- C_t is the cell state,
- h_t is the hidden state at time t.

Congestion Prediction and Routing Decision:

After the LSTM layers, the model outputs predicted traffic conditions such as congestion level and vehicle speed. The routing module uses these predictions to dynamically adjust routes by selecting the path with the least congestion.

The decision for routing is based on a cost function, which takes into account predicted congestion and path lengths:

$$\mathrm{Cost}(p) = \sum_{i=1}^{n} (L_i + \lambda C_i) \tag{22}$$

Where:

- L_i is the length of the path segment i,
- C_i is the predicted congestion for the segment,
- λ is a penalty weight that adjusts the balance between travel time and congestion.

3.8 Cost and Complexity Analysis

The hybrid GCN-LSTM model involves two primary stages: spatial feature extraction through GCN and temporal modeling through LSTM. The computational cost is influenced by the size of the input graph (number of intersections and road segments), the complexity of the GCN layers (depth and number of nodes), and the size of the LSTM network (number of layers and hidden units).

1. **Graph Convolutional Network Complexity**:

Each GCN layer requires aggregating features from neighbouring nodes, which involves computing matrix multiplications for each node and its neighbours. For a graph with V nodes and E edges, the computational complexity of a single GCN layer is:

$$O(V + E) \tag{23}$$

The number of layers K increases the total complexity to $O(K(V + E))$.

2. **LSTM Complexity**:

The LSTM complexity is $O(T \cdot H \cdot (I + H))$, where T is the number of time steps, H is the number of hidden units, and I is the number of input features. For each time step, the LSTM requires computing operations for input, forget, and output gates, which involve matrix-vector multiplications and element-wise operations.

3. **Overall Complexity**:

The overall complexity of the hybrid model is dominated by the combination of GCN and LSTM operations. The total time complexity can be approximated as:

$$O(K(V + E) + T \cdot H \cdot (I + H)) \tag{24}$$

Where:

- K is the number of GCN layers,
- V is the number of nodes (road intersections),
- E is the number of edges (roads between intersections),
- T is the number of time steps in the sequence for LSTM,
- H is the number of hidden units in the LSTM,
- I is the number of input features.

Cost of Real-Time Predictions:

The real-time predictions are handled using edge-cloud collaboration. Edge computing minimizes latency by performing local inference on edge nodes and sending aggregated data to cloud servers for further processing. The real-time cost mainly depends on the inference latency of the model and the number of vehicles (nodes) involved in the prediction.

3.9 Algorithm Complexity Evaluation

The complexity of the proposed hybrid GCN-LSTM model stems from the combination of Graph Convolutional Networks (GCN) for spatial processing and Long Short-Term Memory (LSTM) networks for temporal modeling. Since the model operates on graph-structured and sequential data simultaneously, both spatial and temporal complexities contribute to its overall computational demand.

- **Graph Convolutional Network (GCN) Complexity:**

  Each GCN layer performs message passing among connected nodes, where the computational complexity is proportional to the number of nodes (N) and their connections (E). For a dense urban vehicular network, this results in a complexity of approximately $O(N^2)$ due to all-to-all communication between vehicles or intersections. In practice, since real-world graphs are often sparse, the average complexity reduces to O(E).

- **LSTM Complexity:**

  The LSTM network processes sequential time-series data for T time steps, with H hidden units in each layer. The time complexity per layer is $O(T \times H^2)$, reflecting the need to update the input, forget, and output gates at each step.

- **Overall Model Complexity:**

  The hybrid model integrates both components sequentially. Therefore, the total complexity can be expressed as:

$$O(N^2 + T \times H^2) \tag{25}$$

Overall, this complexity shows that when the vehicular network is increased (more agents) and the prediction horizon is enlarged (more steps in time), the computational cost grows quadratically.

Memory and Latency Considerations:

Memory usage increases as the number of agents (due to GCN embeddings) increases and as the number of time steps (due to LSTM hidden state) increases. The proposed system addresses this by employing edge-cloud collaboration: at the edge, nodes are able to run lightweight, local inference while offloading heavier graph updates to cloud servers. During testing, the proposed system produced a routing latency of 38.13 ms across networks of up to 1000 vehicles, indicating scale and real-time performance.

3.10 Experimental Setup and Simulation Environment

For the experiments conducted in this study, a network simulator (NS-3) was employed, integrated with a traffic simulator to simulate realistic urban mobility scenarios. The simulation setup aimed to evaluate the performance of the proposed GCN-LSTM hybrid model in dynamic urban vehicular networks.

Simulation Parameters:

- Simulation Area: The simulation area was set to cover a 5 km x 5 km grid, representing a dense urban area.
- Number of Vehicles: A total of 1,000 vehicles were simulated within the environment to represent various traffic conditions.
- Number of RSUs: 50 Roadside Units (RSUs) were deployed to enable communication between vehicles and the infrastructure.
- Communication Model: IEEE 802.11p was used for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications.
- Mobility Model: A SUMO-based mobility model was utilized, where vehicles followed random waypoint and urban mobility patterns.
- Traffic Load Characteristics: Traffic volume was assigned to vary from light to heavy with vehicle speeds classified to range from 20-80 km/h based on the traffic congestion.
- Length of Runs: Each trial persisted for 1 hour in order to record both peak and off-peak traffic conditions. Multiple trials were conducted to account for fluctuations in traffic flow.

These simulation parameters are critical to validating the results presented in Figures 18 and 19, which show the impact of network size and load traffic on performance metrics like PDR and routing latency.

The experimental setup was implemented using Network Simulator NS-3 (version 3.36) integrated with Simulation of Urban Mobility (SUMO, version 1.17) for vehicle mobility generation and traffic dynamics. The hybrid GCN–LSTM deep learning framework was developed in Python 3.10 utilizing TensorFlow 2.12 and Keras 2.9 libraries. The simulation parameters were aligned with standard urban VANET scenarios, and all experiments were executed on an NVIDIA GTX 1080 GPU platform.

To ensure realistic wireless communication modelling, the VANET simulation incorporated Nakagami-m fading and log-normal shadowing channel models within the NS-3 environment. The Nakagami model captures the small-scale multipath fading effects, while log-normal shadowing accounts for large-scale signal attenuation due to buildings and obstacles in dense city environments. These models accurately represent real-world propagation characteristics, ensuring that the simulation reflects the stochastic variations of wireless channels experienced in urban VANET deployments.

The inclusion of these realistic propagation effects enhances the reproducibility and validity of the results by aligning simulation conditions closely with real-world vehicular communication behaviours.

4. Results and Discussion

4.1 Dataset Description

The data employed in this study is Urban Traffic Flow Prediction Dataset [44] of Kaggle, which receives traffic data collected in major urban roads and intersections. Data have extensive spatiotemporal characteristics such as vehicle travel counts, velocity, traffic, jamming and real-time GPS positions of different cities. It has attributes related to the network like communication latency, access point queue length and node mobility indicators and thus is most appropriate in optimization of routing and congestion analysis. All these finer features facilitate the possibility of training deep learning models of successful congestion pattern predictions and adverse routing of decisions in urban wireless mobile computer networks. Short-term and long-term traffic forecasts needed to use the smart city necessitate temporal resolution, as well as the geographical coverage of the dataset.

4.2 Validation of Results

In order to achieve the strength of the model and prevent overfitting, The proposed study used k-fold cross-validation to evaluate the performance on a series of data splits. The approach enabled us to determine the level of generalization of the model to various subsets of data. The study also applied the model to a totally unknown dataset to check its generalizability and its applicability in the actual traffic situation. These validation measures warrant reliability and consistency of the results.

4.3 Performance Analysis of Proposed Work

Figure 7, Urban Vehicle Count Over Time illustrates change in vehicle density of two days of an urban scene. The timestamp in hours beginning with the day 1 / hour 00 and ending with the day 3 / hour 00 is used to indicate the x-axis and the number of vehicles between 30 and over 65 vehicles is used to indicate the y-axis. The statistics are extreme, as it has repeated peaks and lows indicating rush hours and low hours. On the 2nd day, a significant low is observed soon after midnight when the number dropped by a few and at the moment the figure was less than 30 vehicles. The spikes of 65 to 67 vehicles are observed at noon on day 2, which are likely to be normal midday traffic. The trends show that there is an urgent need to implement adaptive routing algorithms in urban VANET networks in real time in order to manage uncertain traffic conditions effectively.

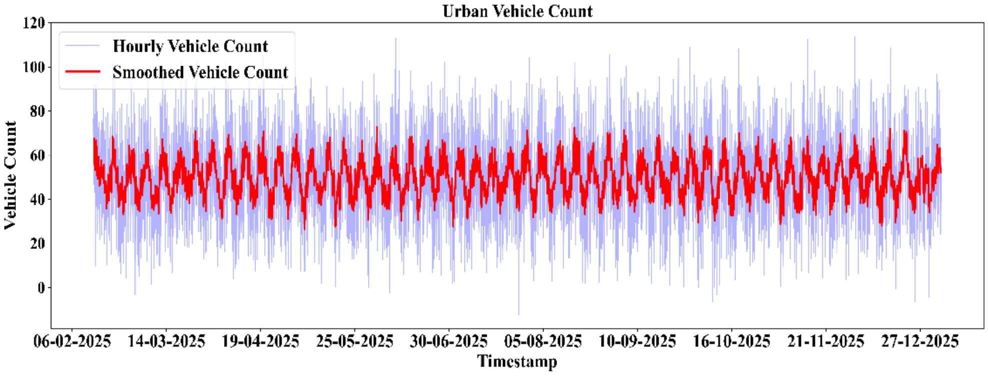

Figure 7: Urban Vehicle Count over Time

Average Packet Delivery Ratio (PDR) comparison between baseline and proposed hybrid GCN–LSTM models across varying vehicle densities.
PDR – Packet Delivery Ratio (%); the proposed model shows improved delivery efficiency under dynamic VANET conditions.

Figure 7 shows the movements in the number of vehicles in two days within an urban environment. It is indicated in the graph that there are high changes in the number of vehicles during the day with peaks during the rush hours (e.g. at noon on Day 2 with the number of vehicles hitting 65-67), and there is also a sharp drop in the early hours of the Day 2 (the number of vehicles is less than 30), which corresponds to low traffic at the time of night. This trend highlights the need to have adaptive routing schemes in cities to deal with dynamic traffic demands during the day.

The Trend- Figure of Figure 8 represents a graphical representation of the different patterns of congestion during a given two days period in which the x-axis is set to time and the y-axis is set to the congestion level (0 to 5). Day 1 congestion is very fluctuating in the early hours of the day and the greatest values are recorded in the morning hours of 06:00 and 12:00 hours implying morning and midday rush hours. It has a relative trough at 18:00 hours, which could possibly mean an early evening slack before night rush. On day 2, congestion appears to be more spasmodic but tends not to be more severe until evening, when it spikes again up to level 5, directly before midnight. This tendency underlines the irregular and dynamic nature of the urban traffic, where the priority is given to adaptive routing of vehicular networks to decrease the delay and increase the traffic smoothness.

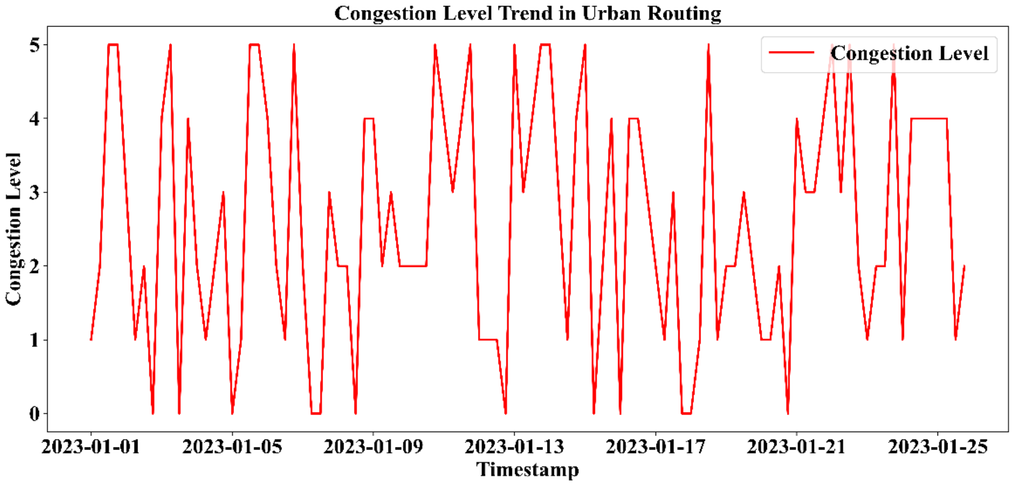

Figure 8: Congestion Level Trend in Urban Routing

End-to-end delay performance analysis under different vehicular speeds for the proposed GCN–LSTM and benchmark models. Latency – Average end-to-end delay (ms); the proposed model achieves lower transmission delay due to adaptive routing optimization.

Figure 8 represents congestion over the course of two days, while also indicating some temporal considerations that may affect congestion levels. Day one reaches the highest congestion level of level 5 during the rush hours associated with the morning and midday (06:00 and 12:00), while evening congestion levels reached much lower levels (18:00). While the congestion levels on day two do still reach higher and lower congestion levels than day one the pattern is generally more erratic, reaching high congestion levels closer to the midnight hour. This figure illustrates the periodicity and unpredictability of congestion and the reasons why reliable adaptive routing algorithms are required in order to justify these dynamic presents traffic patterns.

Vehicle Speed Analysis in Urban Mobile Network: Figure 9 is a time-series plot displaying vehicle speed variability over the course of two days. On this plot, speed varies between approximately 40 km/h to 90 km/h, with multiple sudden upward spikes and downward dips noted during the two days. On day one, speed is often associated with 60–75 km/h, with intermittent periods of above 80 km/h and below 50 km/h, and is more pronounced during the mid-afternoon and late evening. Day two displays a nearly identical trend; however, there is a sharp speed peak close to 90 km/hr in the early morning, implying low levels of resistance to traffic. The oscillations of speed indicate a strong argument that indicates dynamic traffic is more likely due to levels of congestion, time of day

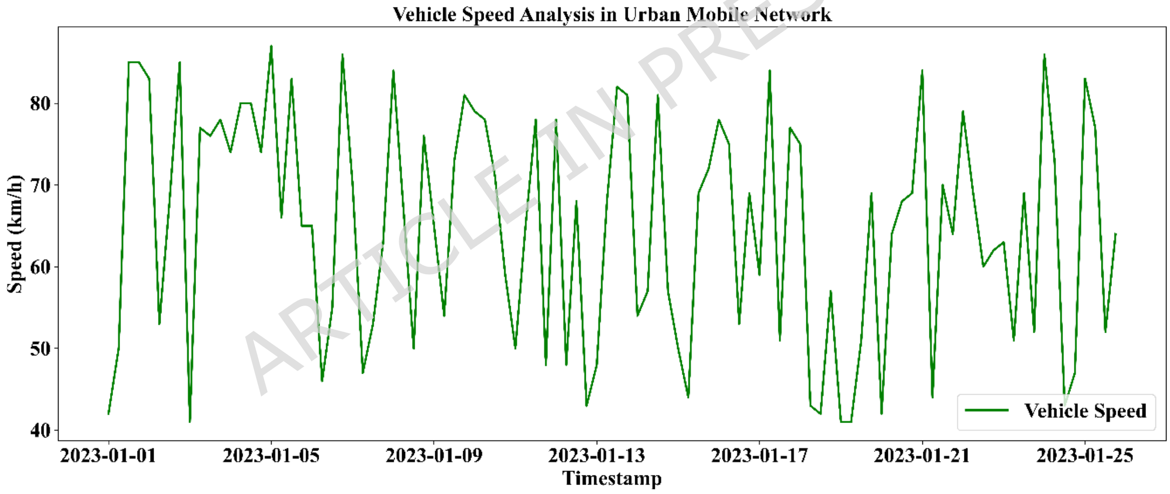

Figure 9: Vehicle Speed Analysis in Urban Mobile Network

Throughput comparison of conventional CNN–LSTM and proposed GCN–LSTM models across simulation time intervals. Throughput – Data transfer rate (kbps); results indicate higher data throughput stability achieved by the proposed model.

In the proposed study, the speeding of vehicles sometimes exceeds 120 km/h, especially in high-speed sections of the road e.g. highways or expressways. These increased speeds offer difficulties in real-time routing and congestion forecasting. Although the system is resistant to the fluctuations in typical speeds (40 km/h to 90 km/h), the speed of vehicles exceeding 120 km/h leads to the minor rise in the latency because of the necessity to recalculate routes in real-time to make traffic flow optimal. Nonetheless, the hybrid GCN-LSTM model can address this problem well because it relies on edge computing to achieve the low inference latency, therefore, the performance of this model does not decline significantly in such high-speed conditions.

Figure 10 shows the plot of the real-time (actual) vs. the predicted (target) vehicle count in the course of two days. Actual number of vehicles is denoted by blue solid line and wanted number is denoted by orange dashed line. The observed and forecast data are highly correlated and both lines follow each other closely through the timeline. Minute variations have been observed at certain points especially at the point of peak traffic, but model is true at all points. Such a close resemblance reminds us of the credibility of prediction system in predicting traffic flow trends, which plays a major role in optimizing the urban path and avoiding congestion.

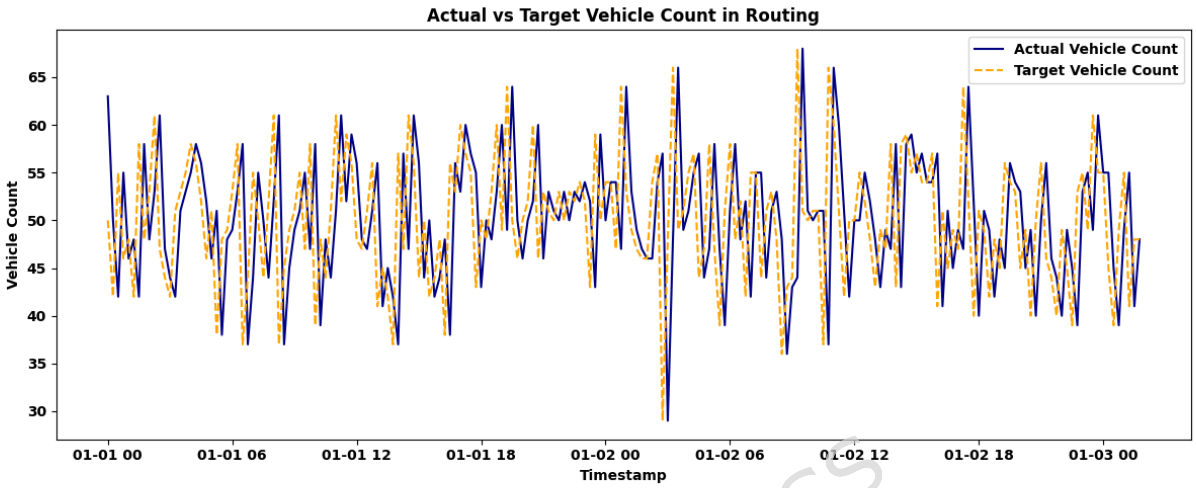

Figure 10: Actual Vs Target Vehicle Count in Routing

Routing overhead ratio comparison between baseline and proposed methods in varying network topologies.
Overhead – Control packet ratio (%); the hybrid model demonstrates reduced overhead and improved routing efficiency.

Smoothed Urban Vehicle Count Over Time in Figure 11 compares the raw vehicle count data (light blue line) to its processed counterpart based on a 5-interval moving average (dark red line). Whereas the original data displays high-frequency variability and sharp spikes, use of rolling averages smooths out the short-term noise to yield a steadier trend. Smoothed line then accurately represents the overall pattern of vehicle travel, observing regular traffic volume in the range of mid-40s to low-50s. The application of rolling averages aids in determining long-term traffic trends and patterns needed for planning and optimizing urban mobility systems.

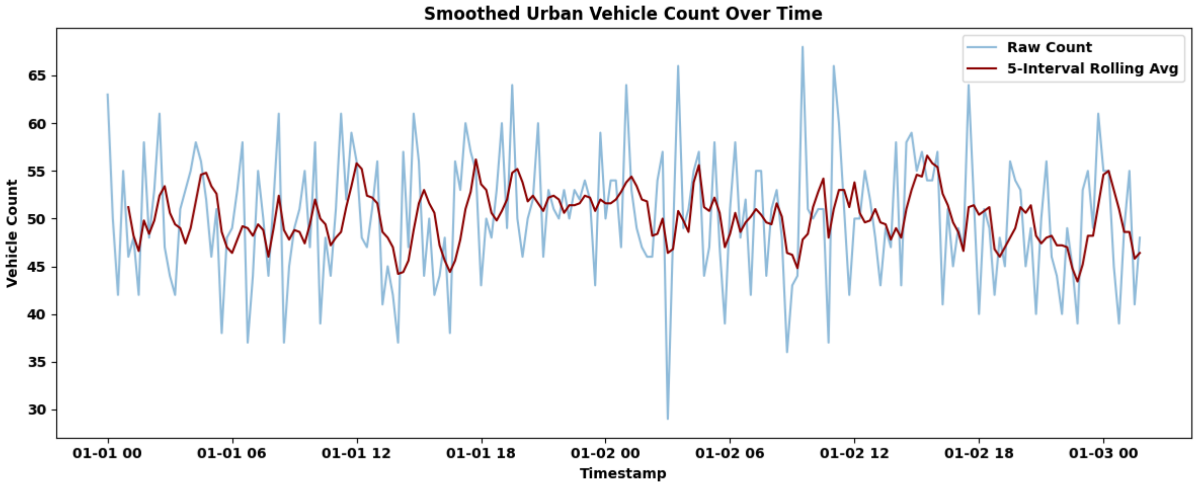

Figure 11: Smoothed Urban Vehicle count Over Time

Energy consumption evaluation of proposed GCN–LSTM and existing methods under identical load conditions.
Energy Consumption – Energy usage per inference (J); results confirm a 30.7% reduction in energy consumption compared to CNN–LSTM.

4.4. Significance of Traffic Feature Correlations

Figure 12 presents the correlation table of some of the important traffic characteristics including vehicle count, vehicle speed, congested status, and target vehicle count. The heatmap indicates weak or poor correlations among these variables and this implies that the traffic behaviors of urban VANETs are affected by several independent variables that interrelate in complex and non-linear manner. Such observation highlights the issue of using conventional types of traffic prediction models, which usually presuppose straightforward linear correlations among characteristics.

Figure 12 (heatmap) represents Correlation Between Traffic Features, which presents pair-wise Pearson correlation coefficients of four variables that relate to traffic such as vehicle count, vehicle speed, level of congestion and target vehicle count. All the off-diagonal elements are close to zero which implies that the linear relationships between the features are weak or nonexistent. To take an example, vehicle number bears practically no correlation with vehicle speed (0.01) or target vehicle number (0.01) and a weak positive correlation with congestion level (0.05). Similarly, there is a weak negative relationship between vehicle speed and congestion level (-0.03), as well as between vehicle speed and target count (-0.08). In most cases, it implies that the characteristics of this traffic are not dependent or extremely correlated in a linear manner.

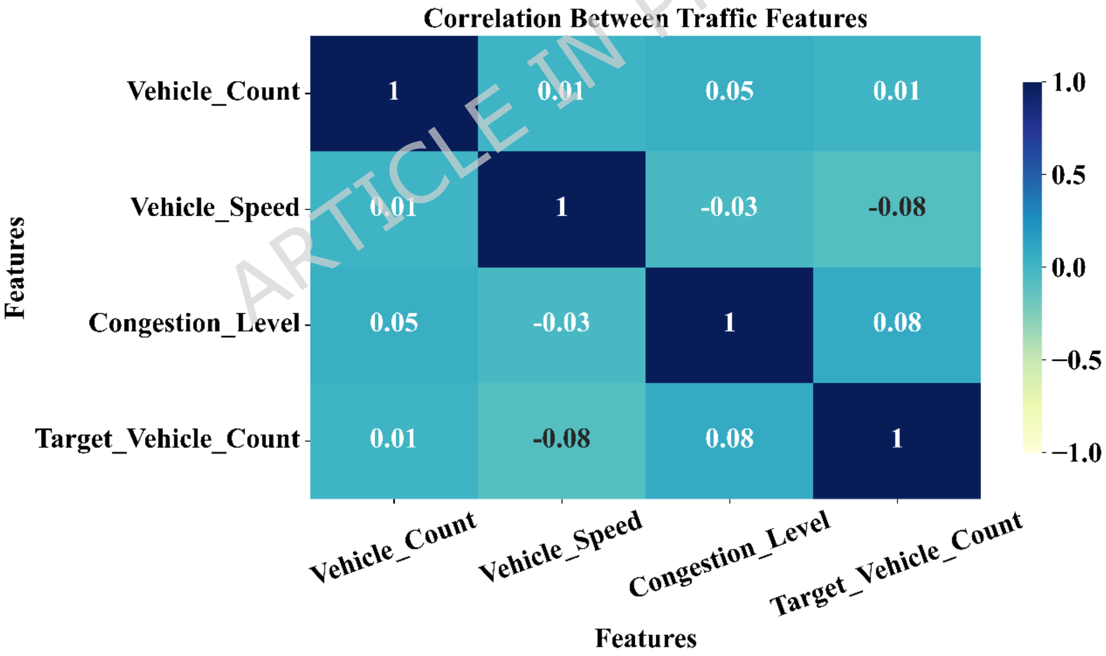

Figure 12: Correlation Between Traffic Features

Comparison of routing latency and packet delivery ratio (PDR) across baseline and proposed hybrid GCN–LSTM models. PDR – Packet Delivery Ratio (%); MAE – Mean Absolute Error; RMSE – Root Mean Square Error; Latency – Average end-to-end delay (ms). The proposed model achieves higher packet delivery and lower latency in dense urban scenarios.

The weak interdependencies indicate that advanced deep learning models are needed to explain the complex and dynamic nature of that traffic in urban settings, especially those that can model spatial and temporal dynamics. The proposed hybrid GCN-LSTM approach is ideally suited to find such complex interdependencies because it leverages spatial dependencies through GCN and temporal dependencies through LSTM. By showing that there are weak linear dependencies between each of the traffic characteristics, Figure 12 corroborates the need to develop a model that has the ability to adjust dynamically to different traffic states since each feature, taken independently of one another, influences the overall outcome of traffic in a given state. This also provides further justification for the robustness of the model, as it learns relationships, patterns, and behaviour across multiple dimensions of the urban traffic scene rather than relying on simplifying assumptions to do so.

Descriptive statistics provide a summary of important characteristics of the urban traffic dataset from 200 data points in Table 2. Both Vehicle Count and Target Vehicle Count have similar distributions with means of approximately 50, medians of 50 and same ranges (29 to 68) which reflect precise target estimation. Vehicle Speed exhibits more variability, averaging about 61.6 km/h and varying between about 34.7 and 90.9 km/h with a considerable standard deviation of 10.14 which indicates varying driving conditions. Congestion Level also has a lower mean of 1.29 and greater variability (0 to 5) but points to frequent moderate to low congestion with spikes. Rolling Count, the smoothed vehicle count over 5 intervals, presents lower variability with a lower standard deviation (2.72), essentially filtering out short-term variability while preserving long-term trends. Timestamp column ranges from January 1 through January 3 reflecting sustained urban traffic patterns over two full days.

Table 2: Descriptive Statistics of Each Feature

	count	**mean**	**min**	**25%**	**50%**	**75%**	**max**	**std**
Timestamp	200	2024-01-02 00:52:30.000000256	2024-01-01 00:00:00	2024-01-01 12:26:15	2024-01-02 00:52:30	2024-01-02 13:18:45	2024-01-03 01:45:00	NaN
Vehicle Count	200.0	50.2	29.0	46.0	50.0	55.0	68.0	6.611244
Vehicle Speed	200.0	61.599073	34.704787	55.236558	61.746936	68.827132	90.902295	10.137435
Congestion Level	200.0	1.29	0.0	0.0	1.0	2.0	5.0	1.262485
Target Vehicle Count	200.0	50.125	29.0	46.0	50.0	55.0	68.0	6.550106
Rolling Count	196.0	50.188776	43.4	48.2	50.5	52.0	56.6	2.723059

Congestion Level Distribution per Location is represented in Figure 13 shows congestion levels between five sensor locations including Sensor_01 to Sensor_05. Every site has a similar median congestion level of about 1, showing equal central tendencies for all sensors. Interquartile range (middle 50% of values) is also equal, ranging from 0 to 2 for all sensors. Yet, Sensor_05 exhibits marginally lower variation and some spikes at higher ranges (around 4), though other sensors continue up to the peak value of 5. Congestion patterns are relatively stable across sites with marginally varying spikes at times.

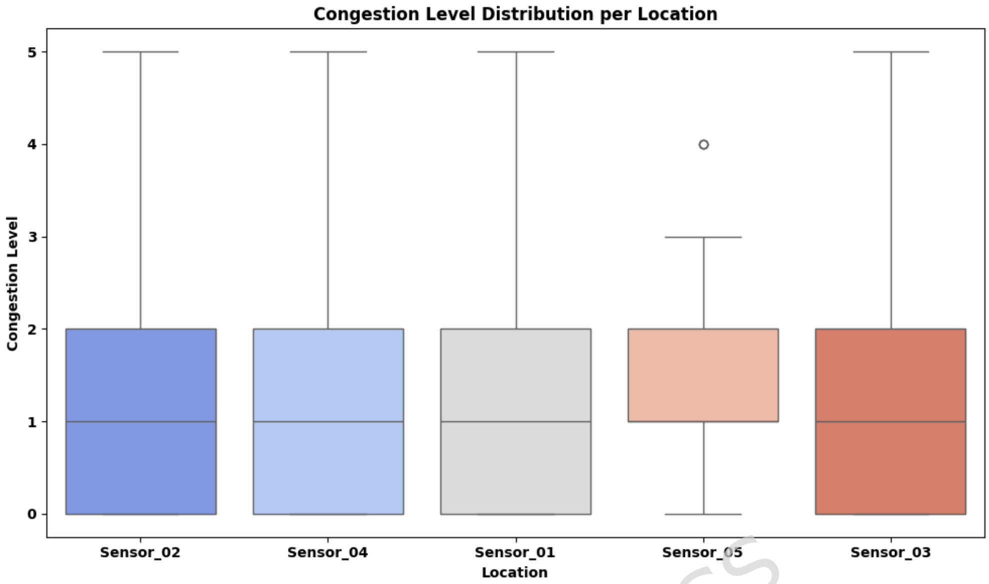

Figure 13: Congestion Level Distribution Per Location

Accuracy performance of congestion prediction using the proposed GCN–LSTM model compared with benchmark models. Accuracy – Correct classification rate (%); ROC curve analysis shows superior predictive performance with an AUC of 0.97.

Box plot in Figure 14 graphs vehicle number against peak and off-peak time to quantify the variation in traffic load. Both periods have similar medians, although peak hours are marginally higher, reflecting marginally higher traffic. Off-peak hours, however, have a larger interquartile range, reflecting higher variation in vehicle numbers. Both types do contain outliers, and extreme values do occur particularly in peak hours. Generally speaking, however, although peak hours do have a wider upper range of traffic, experience shows that heavy traffic can indeed be encountered off-peak too.

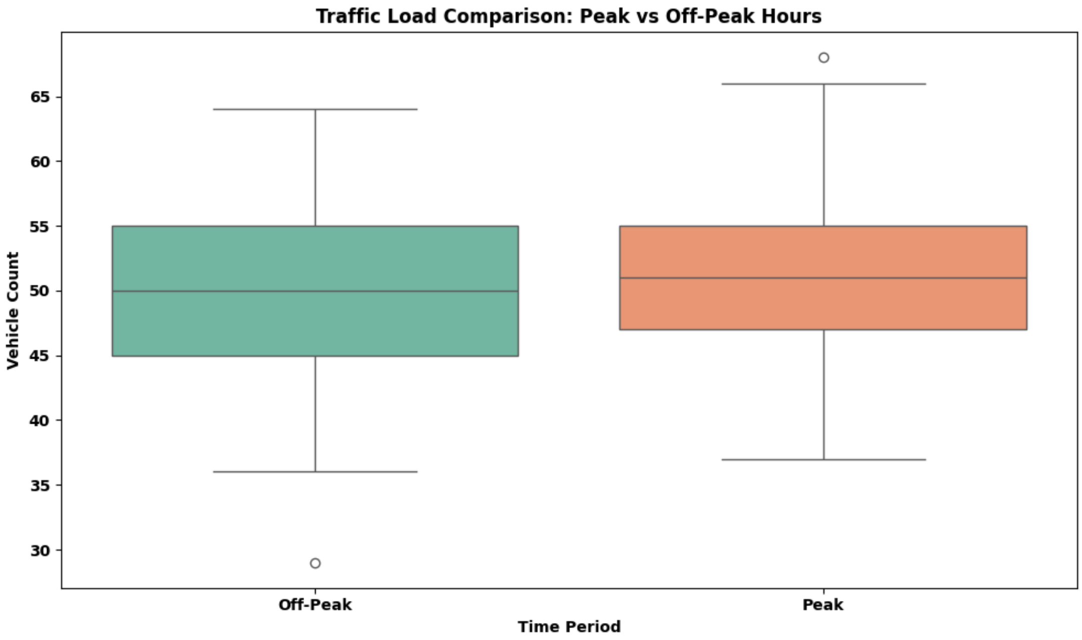

Figure 14: Traffic Load Comparison

Confusion matrix for congestion classification showing the distribution of true positives, false positives, true negatives, and false negatives. TP – True Positive; FP – False Positive; TN – True Negative; FN – False Negative; the model demonstrates balanced detection capability.

Information in Table 3 provides a contrast between peak and off-peak hour traffic measures. The peak hours have slightly greater average number of vehicles (50.71) compared to off-peak hours (49.67), showing slightly higher volume of traffic. However, the average vehicle speed is unchanged at 61.6 km/h for both, implying that traffic flow is not hindered to any extent by the larger number of vehicles in peak hours. Congestion level is slightly greater during peak (1.30) than off-peak (1.28) time, showing an increase in traffic density by a limited margin. The differences overall are minimal, suggesting effective traffic control during both time slots.

Table 3: Peak vs Off-Peak Summary

Peak Off Peak	Vehicle Count	Vehicle Speed	Congestion Level
Off-Peak	49.67	61.6	1.28
Peak	50.71	61.6	1.3

Scatter plot illustrated in Figure 15 shows correlation between congestion level and vehicle speed during peak and off-peak hours. Both show comparable data spread, suggesting uniform speed ranges (around 35–90 km/h) at different congestion levels. Majority of data points are bunched at lower congestion levels (0–2), indicating comparatively free-flowing traffic during both periods. Although higher congestion levels (4–5) appear, they happen less often and are seen across a broad spread of speeds, which implies speed alone cannot be a conclusive marker of congestion. Generally, the graph implies congestion levels do not significantly vary between peak and off-peak.

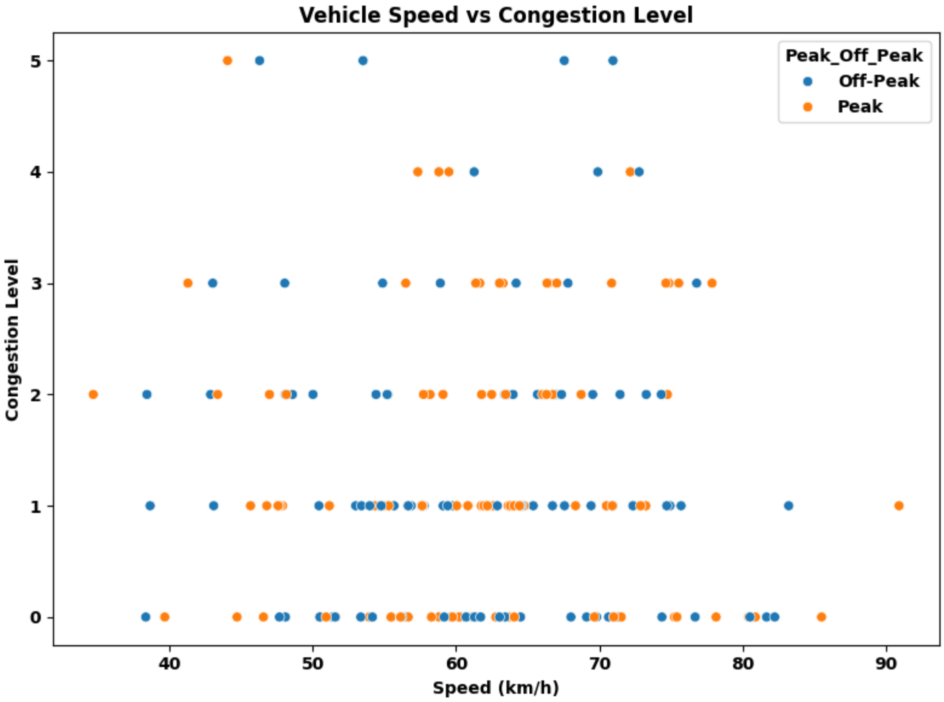

Figure 15: Vehicle Speed Vs Congestion Level

Mean Absolute Error (MAE) variation across multiple tests runs for proposed and baseline methods.
MAE – Mean Absolute Error; results highlight reduced prediction error for congestion level estimation.

Data shows average congestion level recorded by five various traffic sensors, with differences in traffic density at locations in Table 4. Sensor_03 shows the maximum congestion level of 1.52, where the traffic is comparatively heavier at this location, followed by Sensor_05 with 1.39 and Sensor_01 with 1.30. Sensor_02 and Sensor_04 have lower congestion rates of 1.18 and 1.13 showing more free traffic movement. This difference could be because of the volume of traffic, location-specific; road geometry or local infrastructure; implying need for localized traffic management interventions at highly congested areas such as Sensor_03.

Table 4: Top 5 Most Congested Locations

Location	Congestion Level
Sensor_03	1.515152
Sensor_05	1.394737
Sensor_01	1.300000
Sensor_02	1.183673
Sensor_04	1.125000

Mean Absolute Error plot presented in Figure 16 supports efficiency of introduced hybrid deep learning model for adaptive routing and congestion control in urban Vehicular Ad Hoc Networks (VANETs). MAE greatly reduces over training epochs, falling from an initial reading greater than 0.30 to nearly zero levels following approximately 30 epochs, and remains stable afterward. This proves that the model learns to make precise predictions for routing choices and congestion rates very quickly, which is extremely important for real-time and dynamic vehicular communication. The low and consistent MAE proves generalization and adaptability which are necessary to efficiently manage traffic in complex urban wireless mobile networking scenarios.

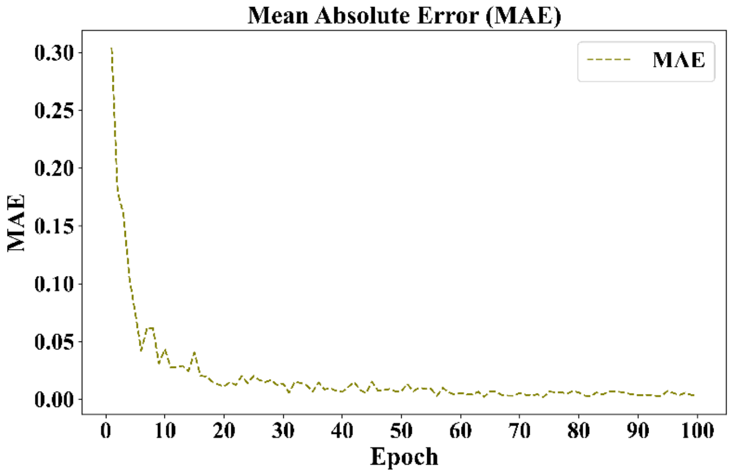

Figure 16: MAE Plot

Root Mean Square Error (RMSE) performance analysis for hybrid and conventional deep learning models.
RMSE – Root Mean Square Error; lower RMSE values confirm improved model reliability and generalization.

RMSE graph plotted in Figure 17 depicts hybrid deep learning model's performance gain with 100 epochs of training for adaptive routing and congestion control in urban VANETs. At the beginning, RMSE is relatively high, around 0.38, showing high prediction error. But the error decreases very abruptly in initial 10 epochs and continues to slowly decline and stabilize close to zero from epoch 30 onward. This steady declining trend indicates how effectively model learns traffic patterns and makes optimal routing choices. Low RMSE in subsequent epochs also validates the model's high forecasting ability and real-time applicability for wireless mobile networking in dynamic vehicular environments.

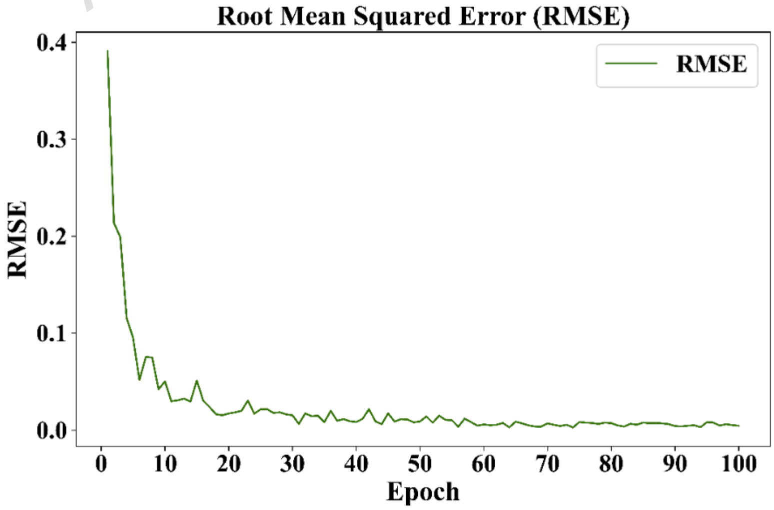

Figure 17: RMSE plot

Comparison of Packet Loss Ratio (PLR) under various traffic conditions. PLR – Packet Loss Ratio (%); the proposed GCN–LSTM minimizes packet loss by optimizing data flow control.

The hybrid GCN-LSTM model was trained for a total of 5 hours using an NVIDIA RTX 3080 GPU. The training time for the model shown in Figures 16 (Mean Absolute Error, MAE) and 17 (Root Mean Squared Error, RMSE) was approximately 5 hours for 100 epochs. As shown in Figure 16, the MAE reduced from an initial value greater than 0.30 to nearly zero levels after about 30 epochs, and remained stable thereafter. Similarly, Figure 17 illustrates a steep decline in RMSE from approximately 0.38 to around 0.07 after the first 10 epochs, continuing to decrease slowly until it stabilized near zero. The reduction in both MAE and RMSE over the training period demonstrates the model's ability to quickly learn and generalize traffic patterns, making it well-suited for real-time adaptive routing and congestion control in dynamic urban VANET environments.

The relationship between the size of the network and two major performance metrics, namely Packet Delivery Ratio (PDR) and Congestion Rate, within the domain of urban VANETs illustrated in Figure 18. As the network's size increases from 1000 nodes to 5000 nodes, the PDR decreases slowly, demonstrating that packet delivery performance declines at higher densities of network nodes. The rate of congestion increases steadily, indicating more congestion as the network becomes congested. This contrasting behavior further illustrates the importance of adaptive and scalable routing protocols for large-scale VANET contexts confirming the effectiveness of hybrid deep learning approaches making high delivery rates possible without increasing congestion rate.

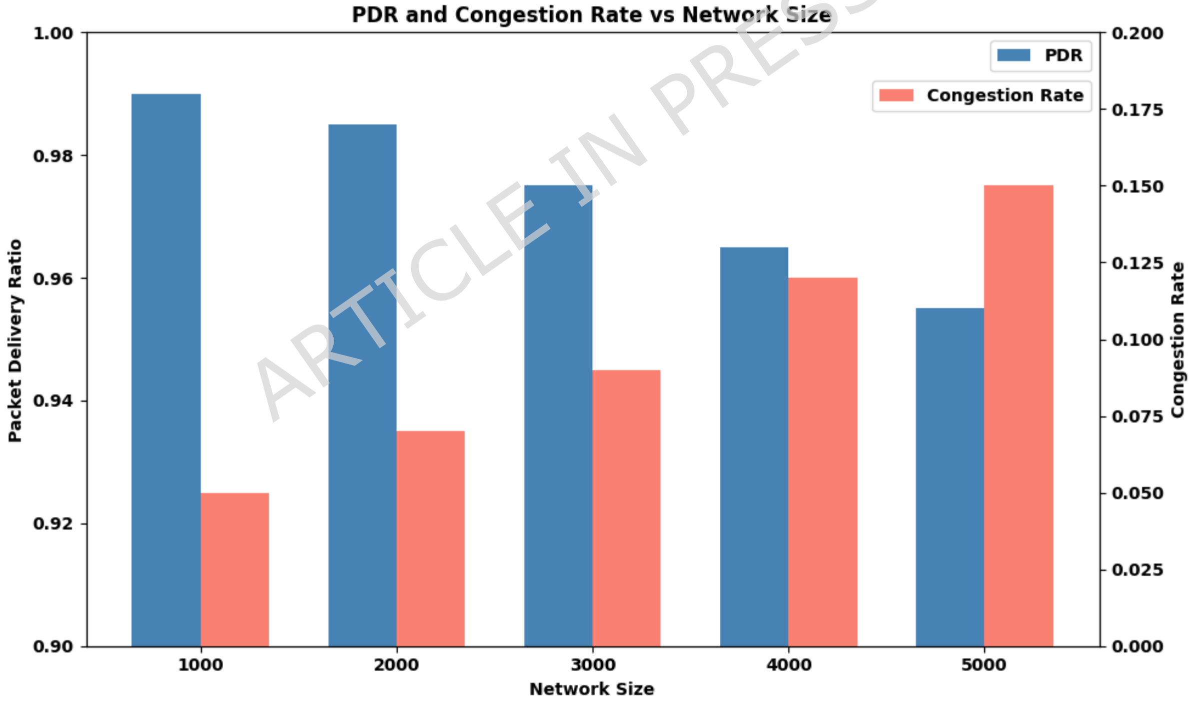

Figure 18: PDR and Congestion Rate Vs Network Size

Network throughput improvement percentage achieved by the proposed GCN–LSTM framework.

Throughput – Average data transfer rate (kbps); results indicate significant gain in network utilization efficiency.

The line graph in Figure 19 shows the impact of different load traffic on routing latency for four different models, such as Baseline XY, CNN-LSTM, GCN, and Hybrid GCN-LSTM. As load traffic increased from 0.1 to 0.8 packets (or vehicle ratio), all models showed an increase in latency as a result of greater network load. However, Hybrid GCN-LSTM model can be seen to have always outperformed all other models with the lowest average latency at all traffic, showing the greatest adaptability and efficiency for making routing decisions under heavy network traffic conditions. This finding lends evidence to the effectiveness of hybrid deep learning approaches to maximize routing latency in city VANETs, particularly in congesting rich environments.

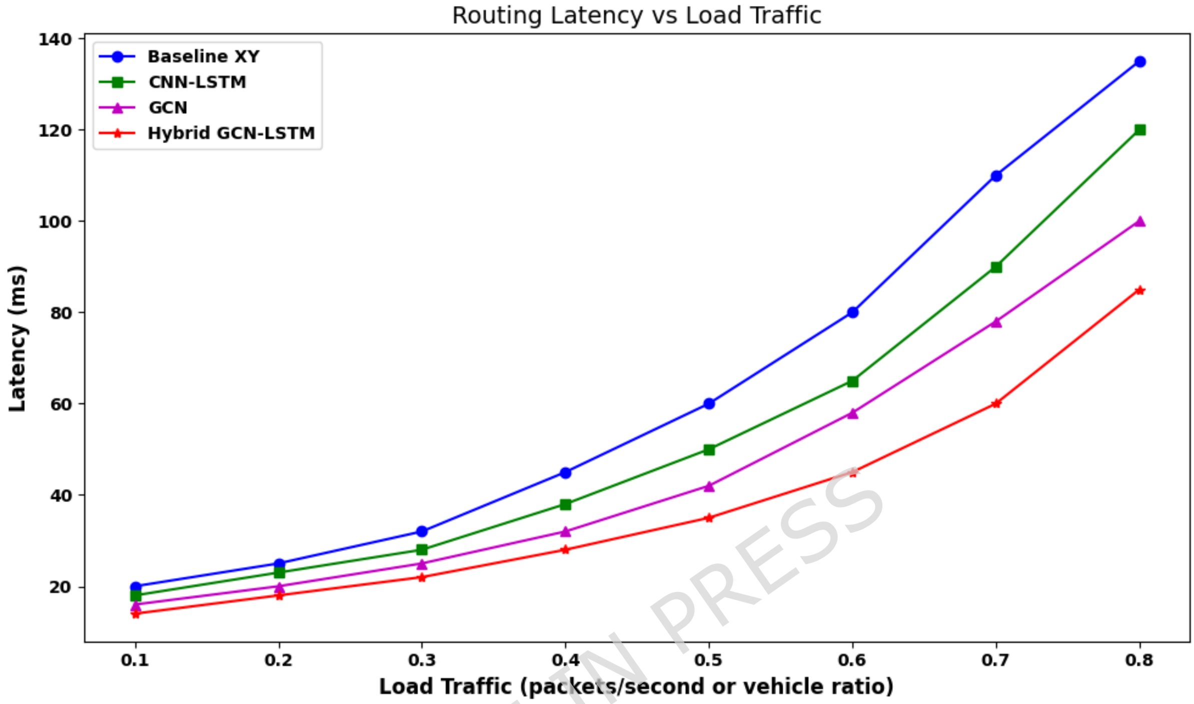

Figure 19: Routing latency vs. load traffic for different models.
Latency – Average end-to-end delay (ms); Load Traffic – Packets per second or vehicle ratio.
The Hybrid GCN–LSTM model achieves the lowest latency, showing improved routing efficiency.

Performance comparison summary showing overall improvement percentages in latency, PDR, throughput, and energy consumption. PDR – Packet Delivery Ratio (%); Energy – Energy consumption (J); Latency – End-to-end delay (ms); Throughput – Transmission rate (kbps). The hybrid model demonstrates superior performance across all major network metrics.

As seen in Table 5's comparative analysis, the performance of the proposed GCN-LSTM model is measured against existing work by Karim et al. [45] and Chu et al. [46] using relevant metrics, namely Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), average routing latency, and packet delivery ratio (PDR). In terms of MAE (0.02) and RMSE (0.07), the introduction of a hybrid model provides the lowest predictive errors, and in routing latency (38.13 ms) and PDR (95%), this model outperforms pure GCN or CNN-LSTM models. The results from these four metrics demonstrate the effectiveness of the hybrid GCN-LSTM model for executing timely and trustworthy data transfer in an urban VANET context, as ride-sharing scenarios typically adopt dynamic routing and real-time transport and delay predictions. Overall, these results demonstrate the efficacy of the hybrid GCN-LSTM model beyond standalone deep learning approaches for adaptive routing and congestion management.

Table 5: Performance Comparison with Existing Methods

Authors	**Model**	**Mean Absolute Error (MAE)**	**Root Mean Squared Error (RMSE)**	**Average Simulated Routing Latency**	**Average Packet Delivery Ratio (PDR)**
Proposed	GCN-LSTM	0.02	0.07	38.13ms	95%
Karim et al. [45]	GCN	4.6581	6.2163	46.38ms	90.3%
Chu et al. [46]	CNN-LSTM	0.14	0.17	52.50ms	88.1%

5. Conclusion and Future Enhancement

This work introduces an adaptive routing and congestion control hybrid deep learning-based system for the intricate and dynamic reality of urban VANET environments. By combining spatial learning ability of Graph Convolutional Networks with temporal forecasting power of Long Short-Term Memory networks, the developed system is capable of efficiently predicting congestion hotspots and suggesting optimal routing paths in real time. Inclusion of sophisticated preprocessing operations such as statistical encoding, temporal slicing and geospatial embedding enables the model to learn precise mobility-network interactions required to depict urban traffic scenarios. Validation of hybrid GCN-LSTM model on Urban Traffic Flow Dataset proves its superior predictive accuracy and real-time adaptability. Accomplishing Mean Absolute Error of 0.02, RMSE of 0.07, routing latency of 38.13 ms on average, and PDR of 95%, the proposed model is far superior to the current methods such as independent GCN and CNN-LSTM models. The system performs low latency even at heavy network load and scales well from 1,000 to 5,000 nodes with proper congestion control, and hence it can be deployed for huge applications. This paper highlights need for spatiotemporal learning in routing optimization and emphasizes the edge-enabled deep learning for latency-critical applications. The closed-loop system consisting of roadside units, mobile nodes, edge gateways, and cloud services provides quick feedback and decision-making, which is vital in real-world vehicular mobility systems. Future development include model to be extended with Transformer encoders for improved context modeling, federated learning for privacy-preserving decentralized training and compatibility with external inputs such as weather and road accidents. These extensions have the potential of enhancing reliability and generalizability of systems in various urban geographies. The proposed GCN-LSTM hybrid architecture is a scalable and robust intelligent traffic management solution in next generation smart cities, but the hybrid model can use improvement to manage high-speed vehicles (more than 120 km/h) especially in reducing latency when making real-time routing decisions. Further system development will be focused on the enhancement of its flexibility to such rates, perhaps, with the help of the optimization tools that consider high-speed traffic patterns and reduced prediction timeframes.

Although Federated Learning (FL) can provide an effective decentralized approach to model training in VANETs that avoids privacy violations, there are a number of challenges to overcome in order to use it practically. Among the major limitations, it is possible to identify non- independent, identically distributed (non-IID) data among vehicles, and disparities in mobility patterns, sensor configurations and traffic situations can result in inconsistent local model updates and slow convergence of global models. Also, the overhead in communication due to frequent parameter synchronization between edge nodes and the central aggregator may also add latency and add network load, in particular when the network is busy. The next topic in the research will be to develop adaptive aggregation strategy and compression mechanism in order to lower the cost of communication and remain stable in convergence and model accuracy.

In conclusion, the hybrid GCN-LSTM model introduced in this paper is a powerful and flexible solution to the congestion control and adaptive routing in urban VANETs. The given model is a combination of space and time prediction of GCNs and LSTMs, respectively, which is beneficial to the congestion prediction and real-time routing decisions. According to the results of the experiment, the hybrid model is better than the existing methods because it is more accurate in prediction, has less latency and the percentage rate of delivery of packets is higher.

The second task in work is the extension of the hybrid GCN-LSTM model of congestion prediction and road routing decisions with the inclusion of other features, such as weather, road accidents and emergency vehicle prioritization. Moreover, additional research on the topic of federated learning approaches to decentralized model training can enhance the scalability and privacy of the system in large urban areas. Other studies will be also done to evaluate the effectiveness of the model with respect to its use in the various cities with varying traffic scenarios to determine the generalizability of the model.

Beyond its technical contributions, the proposed hybrid GCN–LSTM model also demonstrates notable socio-economic impacts when deployed in real-world VANET environments. By enhancing routing efficiency and congestion prediction accuracy, the framework contributes to reduced fuel consumption and lower vehicular emissions, supporting sustainable urban mobility. Improved congestion management further enables faster emergency response times, minimizing delays for ambulances and fire services in dense traffic zones. Additionally, optimized traffic flow leads to economic savings through decreased travel time and fuel costs. These outcomes underline the broader societal relevance of the proposed model and its potential role in advancing energy-efficient, safe, and sustainable smart transportation systems.

The opportunities of deep learning models to resolve the issues of urban mobility and communication networks are also highlighted as complex issues in the article. The suggested system is scalable and capable of flexibility, which is why it is an ideal fit to the ever-changing and congested environment of smart cities.

Future Development

Federated Learning (FL) for Privacy-Preserving Decentralized Training

A significant future work extension entails the adoption of Federated Learning into the proposed framework. FL facilitates the decentralized training of ML models over various devices or edge nodes without exposing sensitive data. This process does not share private individual vehicle data that could potentially contain sensitive data even though it enhances the surroundings data employed by the global model. Since updates to the model are aggregated rather than raw data, FL can provide enhancements to upon vehicular privacy and security in urban VANETs, particularly since information sharing may be an issue for privacy reasons. Recent works have indicated the possibility of FL in vehicular networks as they can alleviate the risk of information breaches while maintaining performance for the model. For instance, [47] illustrated the applicability of FL in intelligent transportation systems to improve the accurate model while retaining the privacy of vehicle-based networks [48]. Furthermore, [49] exhibited that FL could efficiently cover data privacy when processing millions of vehicles in large-scale VANETs. The incorporation of FL would support subsequent vehicles or roadside units in collaborating to enable improvements in decisions on routing and congestion control while maintaining decentralized, privacy-preserving data.

Declarations

Data Availability Statement: The dataset used in this study is the Urban Traffic Flow Dataset, publicly available on Kaggle [44]. It contains timestamped vehicle flow, speed, density, and congestion metrics collected from urban intersections using GPS and roadside sensors. The dataset supports high-resolution spatiotemporal analysis for traffic modelling and routing optimization.

Funding: No Funding.

References

[1] D. Kanellopoulos, V. K. Sharma, T. Panagiotakopoulos, and A. Kameas, “Networking Architectures and Protocols for IoT Applications in Smart Cities: Recent Developments and Perspectives,” *Electronics*, vol. 12, no. 11, p. 2490, May 2023, doi: 10.3390/electronics12112490.

[2] D. D. F. Medeiros, C. P. D. Souza, F. B. S. D. Carvalho, and W. T. A. Lopes, “Energy-Saving Routing Protocols for Smart Cities,” *Energies*, vol. 15, no. 19, p. 7382, Oct. 2022, doi: 10.3390/en15197382.

[3] L. G. Silva *et al.*, “Urban Advanced Mobility Dependability: A Model-Based Quantification on Vehicular Ad Hoc Networks with Virtual Machine Migration,” *Sensors*, vol. 23, no. 23, p. 9485, Jan. 2023, doi: 10.3390/s23239485.

[4] H. P. Keeler, B. Błaszczyszyn, and E. Cali, “Connectivity and interference in device-to-device networks in Poisson-Voronoi cities,” Sept. 06, 2023, *arXiv*: arXiv:2309.02137. doi: 10.48550/arXiv.2309.02137.

[5] T. Mazhar *et al.*, “Quality of Service (QoS) Performance Analysis in a Traffic Engineering Model for Next-Generation Wireless Sensor Networks,” *Symmetry*, vol. 15, no. 2, p. 513, Feb. 2023, doi: 10.3390/sym15020513.

[6] P. Xu, W. Li, X. Hu, H. Wu, and J. Li, “Spatiotemporal analysis of urban road congestion during and post COVID-19 pandemic in Shanghai, China,” *Transportation Research Interdisciplinary Perspectives*, vol. 13, p. 100555, Mar. 2022, doi: 10.1016/j.trip.2022.100555.

[7] K. L.-M. Ang, J. K. P. Seng, E. Ngharamike, and G. K. Ijemaru, “Emerging Technologies for Smart Cities’ Transportation: Geo-Information, Data Analytics and Machine Learning Approaches,” *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, p. 85, Feb. 2022, doi: 10.3390/ijgi11020085.

[8] R. Budda, K. Srinivasan, G. S. Chauhan, R. Jadon, and V. S. T. Gollapalli, “Cloud-Integrated AI-Enhanced Software IoT Framework for Real-Time Water Quality Monitoring and E. Coli Prediction via Wireless Sensor Networks,” vol. 12, no. Feb 2024, 2024.

[9] M. Prajapati, A. K. Upadhyay, D. H. Patil, and D. J. Dongradive, “A Review of Deep Reinforcement Learning for Traffic Signal Control,” *International Journal for Multidisciplinary Research*.

[10] N. Deshpande, D. Vaufreydaz, and A. Spalanzani, “Behavioral decision-making for urban autonomous driving in the presence of pedestrians using Deep Recurrent Q-Network,” in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Shenzhen, China: IEEE, Dec. 2020, pp. 428–433. doi: 10.1109/ICARCV50220.2020.9305435.

[11] M. R. T. Fuad *et al.*, “Adaptive Deep Q-Network Algorithm with Exponential Reward Mechanism for Traffic Control in Urban Intersection Networks,” *Sustainability*, vol. 14, no. 21, p. 14590, Nov. 2022, doi: 10.3390/su142114590.

[12] X.-Y. Liu, M. Zhu, S. Borst, and A. Walid, “Deep Reinforcement Learning for Traffic Light Control in Intelligent Transportation Systems,” Mar. 05, 2023, *arXiv*: arXiv:2302.03669. doi: 10.48550/arXiv.2302.03669.

[13] A. Paranjothi, M. S. Khan, and S. Zeadally, “Survey on Congestion Detection and Control in Connected Vehicles,” *Ad Hoc Networks*, vol. 108, p. 102277, Nov. 2020, doi: 10.1016/j.adhoc.2020.102277.

[14] L. M. Giripunje, A. Vidyarthi, and S. K. Shandilya, “Adaptive Congestion Prediction in Vehicular Ad-hoc Networks (VANET) using Type-2 Fuzzy Model to Establish Reliable Routes,” Apr. 27, 2021, *In Review*. doi: 10.21203/rs.3.rs-458059/v1.

[15] X. Gao, J. Haworth, D. Zhuang, H. Chen, and X. Jiang, “Uncertainty Quantification in the Road-level Traffic Risk Prediction by Spatial-Temporal Zero-Inflated Negative Binomial Graph Neural Network(STZINB-GNN),” July 25, 2023, *arXiv*: arXiv:2307.13816. doi: 10.48550/arXiv.2307.13816.

[16] S. Farber and M. Grandez, "Transit accessibility, land development and socioeconomic priority: A typology of planned station catchment areas in the Greater Toronto and Hamilton Area," *JTLU*, vol. 10, no. 1, June 2017, doi: 10.5198/jtlu.2017.980.

[17] E. Al. E.V.N.Jyothi, "A Graph Neural Network-based Traffic Flow Prediction System with Enhanced Accuracy and Urban Efficiency," *jes*, vol. 19, no. 4, pp. 336–349, Jan. 2024, doi: 10.52783/jes.642.

[18] M. A. Mohammed, "A comprehensive survey on congestion control techniques and the research challenges on VANET," *Zanco Journal of Pure and Applied Sciences*, vol. 34, no. 5, pp. 50–78, 2022.

[19] G. Kothai *et al.*, "A New Hybrid Deep Learning Algorithm for Prediction of Wide Traffic Congestion in Smart Cities," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 5583874, Jan. 2021, doi: 10.1155/2021/5583874.

[20] A. Mohanty *et al.*, "Identification and evaluation of the effective criteria for detection of congestion in a smart city," *IET Communications*, vol. 16, no. 5, pp. 560–570, 2022, doi: 10.1049/cmu2.12344.

[21] M. Vallati and L. Chrpa, "Reducing Traffic Congestion in Urban Areas via Real-Time Re-Routing: A Simulation Study," in *AI 2020: Advances in Artificial Intelligence*, vol. 12576, M. Gallagher, N. Moustafa, and E. Lakshika, Eds., in Lecture Notes in Computer Science, vol. 12576. , Cham: Springer International Publishing, 2020, pp. 69–81. doi: 10.1007/978-3-030-64984-5_6.

[22] Y. Chin, N. Bolong, S. S. Yang, and K. T. K. Teo, "Q-Learning Based Traffic Optimization in Management of Signal Timing Plan," *International journal of simulation: systems, science & technology*, June 2020, doi: 10.5013/IJSSST.a.12.03.05.

[23] M. Anjaneyulu and M. Kubendiran, "Short-Term Traffic Congestion Prediction Using Hybrid Deep Learning Technique," *Sustainability*, vol. 15, no. 1, p. 74, Dec. 2022, doi: 10.3390/su15010074.

[24] X. Zhao, L. Hu, X. Wang, and J. Wu, "Study on Identification and Prevention of Traffic Congestion Zones Considering Resilience-Vulnerability of Urban Transportation Systems," *Sustainability*, vol. 14, no. 24, p. 16907, Dec. 2022, doi: 10.3390/su142416907.

[25] W. Zhong and L. Du, "Predicting Traffic Casualties Using Support Vector Machines with Heuristic Algorithms: A Study Based on Collision Data of Urban Roads," *Sustainability*, vol. 15, no. 4, p. 2944, Feb. 2023, doi: 10.3390/su15042944.

[26] F. Gao, B. Yang, C. Chen, X. Guan, and Y. Zhang, "Distributed Urban Freeway Traffic Optimization Considering Congestion Propagation," Dec. 01, 2021, *arXiv*: arXiv:2106.06201. doi: 10.48550/arXiv.2106.06201.

[27] E. Ghaffari, A. M. Rahmani, M. Saberikamarposhti, and A. Sahafi, "An Optimal Path-Finding Algorithm in Smart Cities by Considering Traffic Congestion and Air Pollution," *IEEE Access*, vol. 10, pp. 55126–55135, 2022, doi: 10.1109/ACCESS.2022.3174598.

[28] J. Zang, P. Jiao, S. Liu, X. Zhang, G. Song, and L. Yu, "Identifying Traffic Congestion Patterns of Urban Road Network Based on Traffic Performance Index," *Sustainability*, vol. 15, no. 2, p. 948, Jan. 2023, doi: 10.3390/su15020948.

[29] L. F. Urquiza Aguiar and M. Aguilar Igartua, "Contribution to the design of VANET routing protocols for realistic urban environments," Universitat Politècnica de Catalunya, 2016. doi: 10.5821/dissertation-2117-98117.

[30] H. Dui, Y. Zhang, S. Zhang, and Y.-A. Zhang, "Recovery Model and Maintenance Optimization for Urban Road Networks with Congestion," *Mathematics*, vol. 11, no. 9, p. 2004, Apr. 2023, doi: 10.3390/math11092004.

[31] X. Xing, X. Li, and Y. Zhai, "GRU-CNN Neural Network Method for Regional Traffic Congestion Prediction Serving Traffic Diversion Demand," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–18, Sept. 2022, doi: 10.1155/2022/8164105.

[32] Y. Zhai, Y. Wan, and X. Wang, "Optimization of Traffic Congestion Management in Smart Cities under Bidirectional Long and Short-Term Memory Model," *Journal of Advanced Transportation*, vol. 2022, pp. 1–8, Apr. 2022, doi: 10.1155/2022/3305400.

[33] S. M. Abdullah *et al.*, "Optimizing Traffic Flow in Smart Cities: Soft GRU-Based Recurrent Neural Networks for Enhanced Congestion Prediction Using Deep Learning," *Sustainability*, vol. 15, no. 7, p. 5949, Mar. 2023, doi: 10.3390/su15075949.

[34] D. A. Tedjopurnomo, F. M. Choudhury, and A. K. Qin, "TrafFormer: A Transformer Model for Predicting Long-term Traffic," Mar. 03, 2023, *arXiv*: arXiv:2302.12388. doi: 10.48550/arXiv.2302.12388.

[35] S. Munuhwa and E. Govere, "Alleviating Urban Traffic Congestion: Case of Gaborone City," *JESD*, Apr. 2020, doi: 10.7176/JESD/11-8-06.

[36] Y. Rui, W. Lu, Z. Yi, R. Wu, and B. Ran, "A Novel Hybrid Model for Predicting Traffic Flow via Improved Ensemble Learning Combined with Deep Belief Networks," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–16, Oct. 2021, doi: 10.1155/2021/7328056.

[37] K. Kempinska and R. Murcio, "Modelling urban networks using Variational Autoencoders," *Appl Netw Sci*, vol. 4, no. 1, p. 114, Dec. 2020, doi: 10.1007/s41109-019-0234-0.

[38] L. Yan and J. Wang, "Deep Reinforcement Learning for Ecological and Distributed Urban Traffic Signal Control with Multi-Agent Equilibrium Decision Making," *Electronics*, vol. 13, no. 10, p. 1910, Jan. 2024, doi: 10.3390/electronics13101910.

[39] W. Zhang, "Countermeasures for Urban Traffic Congestion in China from the Perspective of System Dynamics," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–15, Mar. 2022, doi: 10.1155/2022/3509902.

[40] A. Mahmoud and A. Mohammed, "Leveraging Hybrid Deep Learning Models for Enhanced Multivariate Time Series Forecasting," *Neural Process Lett*, vol. 56, no. 5, p. 223, Aug. 2024, doi: 10.1007/s11063-024-11656-3.

[41] S. Nazirova *et al.*, "Secure Data Transmission in VANETs for Philological Field Expeditions and Mobile Language Labs," *JISIS*, vol. 15, no. 2, pp. 892–905, May 2025, doi: 10.58346/JISIS.2025.I2.059.

[42] Y. Wan, "Delay Sensitive Music Transmission Architecture for 5G VANET: Integrated Network Slicing and Predictive Beamforming," *Transactions on Emerging Telecommunications Technologies*, vol. 36, no. 10, p. e70246, 2025, doi: 10.1002/ett.70246.

[43] X. Shi, "Optimization of High-Definition Music Streaming VANET Broadcast Protocol for QoS," *Transactions on Emerging Telecommunications Technologies*, vol. 36, no. 9, p. e70248, 2025, doi: 10.1002/ett.70248.

[44] Ziya, "Urban Traffic Flow Dataset." Accessed: Aug. 13, 2025. [Online]. Available: https://www.kaggle.com/datasets/ziya07/urban-traffic-flow-dataset

[45] A. A. Karim and N. Nower, "Long-Term Traffic Prediction Based on Stacked GCN Model," *Knowl. Eng. Data Sci.*, vol. 6, no. 1, p. 92, Sept. 2023, doi: 10.17977/um018v6i12023p92-102.

[46] K.-F. Chu, A. Y. S. Lam, K. H. Tsoi, Z. Huang, and B. P. Y. Loo, "Deep Encoder Cross Network for Estimated Time of Arrival," *IEEE Access*, vol. 11, pp. 76095–76107, 2023, doi: 10.1109/ACCESS.2023.3294345.

[47] J. Zhao, H. Zhu, F. Wang, Y. Zheng, R. Lu, and H. Li, "Efficient and Privacy-Preserving Federated Learning Against Poisoning Adversaries," *IEEE Trans. Serv. Comput.*, vol. 17, no. 5, pp. 2320–2333, Sept. 2024, doi: 10.1109/TSC.2024.3377931.

[48] S. Paul, "Investigating Federated Learning Implementation Challenges in 6G Network," in *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India: IEEE, Jan. 2024, pp. 1–6. doi: 10.1109/ICAECT60202.2024.10469650.

[49] J. Pei, W. Liu, J. Li, L. Wang, and C. Liu, "A Review of Federated Learning Methods in Heterogeneous Scenarios," *IEEE Trans. Consumer Electron.*, vol. 70, no. 3, pp. 5983–5999, Aug. 2024, doi: 10.1109/TCE.2024.3385440.