

Article

A Graph Neural Network (GNN)-Based Approach for Real-Time Estimation of Traffic Speed in Sustainable Smart Cities

Amit Sharma ^{1,2}, Ashutosh Sharma ^{2,3,*}, Polina Nikashina ¹ , Vadim Gavrilenko ¹ , Alexey Tselykh ¹ , Alexander Bozhenyuk ¹, Mehedi Masud ^{4,*}  and Hossam Meshref ⁴

- ¹ Institute of Computer Technology and Information Security, Southern Federal University, Taganrog 347900, Russia; sharma@sfedu.ru (A.S.); nikashina@sfedu.ru (P.N.); vagavrilenko@sfedu.ru (V.G.); tselykh@sfedu.ru (A.T.); avbozhenyuk@sfedu.ru (A.B.)
- ² Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, India
- ³ School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India
- ⁴ Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. 14 Box 11099, Taif 21944, Saudi Arabia; hmeshref@tu.edu.sa
- * Correspondence: sharmaashutosh1326@gmail.com (A.S.); mmasud@tu.edu.sa (M.M.)

Abstract: Planning effective routes and monitoring vehicle traffic are essential for creating sustainable smart cities. Accurate speed prediction is a key component of these efforts, as it aids in alleviating traffic congestion. While their physical proximity is important, the interconnection of these road segments is what significantly contributes to the increase of traffic congestion. This interconnectedness poses a significant challenge to increasing prediction accuracy. To address this, we propose a novel approach based on Deep Graph Neural Networks (DGNNs), which represent the connectedness of road sections as a graph using Graph Neural Networks (GNNs). In this study, we implement the proposed approach, called STGGAN, for real-time traffic-speed estimation using two different actual traffic datasets: PeMSD4 and PeMSD8. The experimental results validate the prediction accuracy values of 96.67% and 98.75% for the PeMSD4 and PeMSD8 datasets, respectively. The computation of mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) also shows a progressive decline in these error values with increasing iteration count, demonstrating the success of the suggested technique. To confirm the feasibility, reliability, and applicability of the suggested STGGAN technique, we also perform a comparison analysis, including several statistical, analytical, and machine-learning- and deep-learning-based approaches. Our work contributes significantly to the field of traffic-speed estimation by considering the structure and characteristics of road networks through the implementation of DGNNs. The proposed technique trains a neural network to accurately predict traffic flow using data from the entire road network. Additionally, we extend DGNNs by incorporating Gated Graph Attention Network (GGAN) blocks, enabling the modification of the input and output to sequential graphs. The prediction accuracy of the proposed model based on DGNNs is thoroughly evaluated through extensive tests on real-world datasets, providing a comprehensive comparison with existing state-of-the-art models for traffic-flow forecasting.

Keywords: Graph Neural Networks (GNNs); traffic-speed prediction; Convolutional Neural Networks (CNNs); smart cities



Citation: Sharma, A.; Sharma, A.; Nikashina, P.; Gavrilenko, V.; Tselykh, A.; Bozhenyuk, A.; Masud, M.; Meshref, H. A Graph Neural Network (GNN)-Based Approach for Real-Time Estimation of Traffic Speed in Sustainable Smart Cities. *Sustainability* **2023**, *15*, 11893. <https://doi.org/10.3390/su151511893>

Academic Editor: Giulio Erberto Cantarella

Received: 11 May 2023

Revised: 5 July 2023

Accepted: 10 July 2023

Published: 2 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The prediction of traffic speed in real time for sustainable smart cities is an essential component for intelligent and efficient transportation systems. Several popular applications of transportation like congestion avoidance, route navigation, and traffic monitoring are deeply reliant on high-quality predictions [1]. However, there exist some complications

in accurate predictions in terms of unbalanced flow patterns and complex dynamic traffic environments. These factors make the prediction system more complex in terms of accuracy. Therefore, numerous studies have attempted to capture these different factors and patterns to improve the prediction accuracy in order to avoid congestions and collisions [2]. Recently, the implementation of deep-learning-based models has achieved exceptional performances in several applications [3]. There are many studies which emphasise the implementation of Deep Neural Network (DNN)-based models for the accurate prediction of traffic-information-related problems. It was found from the survey that most of these studies are based on RNNs (Recurrent Neural Networks) [4], CNNs (Convolutional Neural Networks) [5], or a combination of these two models [6]. The Convolutional Neural Network-based models usually frame the highway network as an image and establish the network-wide traffic flows on the basis of spatial dependency [7]. In comparison with CNN-based models, the Recurrent Neural Network-based models are highly implemented for studying temporal dynamics. However, more efficient traffic prediction outcomes have been achieved by implementing LSTM (long short-term memory) neural networks that are proficient in learning both long- and short-term time series. However, conventional CNNs and RNNs do not explore important characteristics such as road network topology, essential properties of road sectors, and the relationships among them, which can be characterised as a graph. The connection of these road segments inside the structure of the road network is the real cause of the rise in traffic congestion, even though geographical proximity appears to have an impact on the correlation among traffic environments.

In Figure 1, the main road connecting a city with its outskirts is depicted, with vehicles C1, C2, C3, C4, and C5 signifying the traffic flow situated on it. C1, C2, C4, and C5 travel from the city centre towards the outskirts, while C3 moves in the opposite direction. As time passes, during the evening hours, traffic congestion initiates at C1, as shown in Figure 1a. Subsequently, other vehicles, such as C2, C4, and C5, are also affected by this congestion, as illustrated in Figure 1b. Eventually, all vehicles connected to C1 experience heavy congestion, while C3 remains congestion-free, as depicted in Figure 1c. These observations indicate that the spread of traffic situations is primarily influenced by the interconnectedness of road segments in the network structure, rather than by their spatial proximity.

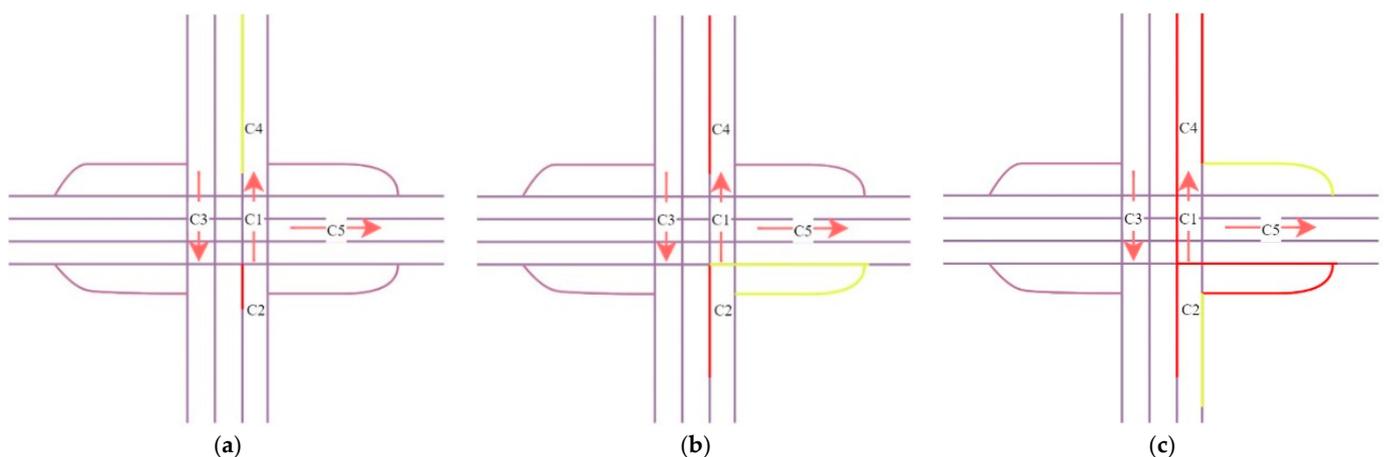


Figure 1. Progression of traffic conditions (a) when traffic congestion begins at C1. (b) After some time, other vehicles C2, C4, and C5 also become affected when (c) vehicles connected to C1 become involved in heavy congestion.

Note: In above Figure, the arrows represent movement of vehicles in a particular lane, red lines represent the traffic congestion and green line represents the smooth passage or slip roads for traffic movement.

In sustainable smart cities, the efficient management of traffic is crucial to optimising transportation systems, reducing congestion, and minimising environmental impacts. Real-time estimation of traffic speed plays a vital role in achieving these goals. Traditional methods of traffic-speed estimation rely on sensors and data collected from fixed locations, which often have limitations in terms of coverage, accuracy, and real-time updates. To overcome these limitations, a Graph Neural Network (GNN)-based approach has emerged as a promising solution for real-time traffic-speed estimation in sustainable smart cities. GNNs are a type of deep-learning model specifically designed to analyse and process graph-structured data, where the nodes represent entities (such as intersections or road segments) and edges represent the relationships or connections between them. The GNN-based approach for traffic-speed estimation leverages the inherent graph structure of urban road networks to capture the spatial dependencies and temporal dynamics of traffic flow.

By considering the interdependencies between different road segments and their influence on each other, GNNs can learn complex patterns and make accurate predictions. In order to accurately predict the traffic speed, several studies on GNNs (Graph Neural Networks) [8,9] and GCNNs (Graph CNNs) [10] have generated novel concepts. This is the inspiration behind this study, where the problem is formulated considering both input and output as a sequential graph. There are still some open research challenges that need to be addressed. The first challenge is how to extract important characteristics from intricate road networks in order to discover patterns or trends in the growth of traffic speeds. The second challenge is that the current research on GNNs only produces a single value and not a series of graphs when the network is fed sequential inputs. The third challenge is that the typical structure of sequence-to-sequence models (Seq2Seq) of the encoder and decoder techniques are inadequate for directly processing graph-structure data.

1.1. Research Contribution

In this study, a brand-new deep Graph Neural Network-based traffic-speed estimate model is suggested. The main contributions of this study are summarised as follows:

1. The structure and characteristics of road networks are taken into consideration by implementing DGNNs. The proposed technique trains a neural network for the accurate prediction of traffic flow using data from the entire road network.
2. The proposed approach is an extension of a DGNN that uses Gated Graph Attention Network (GGAN) blocks for the modification of inputs and outputs to sequential graphs.
3. Extensive experiments on real-world datasets show that the proposed model based on DGNNs predicts outcomes accurately. The suggested model's validation and forecasting of traffic flow are put to the test against current state-of-the-art models.

1.2. Article Organisation

The rest of the article is structured as follows: A few relevant studies are discussed in Section 2. The preliminaries are discussed in Section 3, followed by a description of the proposed model in Section 4. The experimental analysis and performance evaluation of the proposed model is presented in Section 5. Last, the concluding remarks are presented in Section 6, with some suggestions for future research.

2. Related Work

This section discusses a variety of cutting-edge studies on neural networks, Deep Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks that are used to estimate traffic speed.

2.1. Prediction of City Road-Traffic Speed

Researchers from several parts of the world are attracted towards the challenge of predicting or estimating traffic speed on city roads, and several approaches and solutions for the traffic congestion problem have been proposed. Initially, the prediction of traffic speed on roads was considered to be the problem of a time-series sequence that could be

handled using autoregressive moving-average techniques [11]. There are several other approaches, such as gradient boosting [12], k-nearest neighbour [13], and SVR (support vector regression) [14], which are referred to as statistical learning techniques. Statistical learning techniques are more advantageous over time-series techniques because of their capability in aggregating relevant information, which ultimately leads to improvement in prediction performance. However, due to their inadequate representation of complicated scenarios, the potential of these models for performance improvement is severely limited. In recent years, there have been numerous advances in the theoretical aspects underlying Deep Neural Networks as well as effective implementations to different applications on the basis of these studies. The diverse range of Deep Neural Network models have attracted several researchers and received much of attention, particularly CNNs (Convolutional Neural Networks) and RNNs (Recurrent Neural Networks). The CNN models were initially primarily employed to process image data, although since then, they have been implemented for various types of data that can be described as two-dimensional matrices. A large number of convolution kernels that have shared weights make up a Convolutional Neural Network of convolutional layers. A convolution kernel only links with variables in a specified area, dramatically reducing the number of hyperparameters in the neural network. Additionally, in Convolutional Neural Networks, pooling is frequently employed. As a sample scenario, finding the matrix element with the highest value is referred to as maximum pooling. The Recurrent Neural Network consists of a loop structure that enables the model to process data in a consistent form, which includes several forms of sequential data formats like articles and voice. LSTM (long short-term memory) is a type of RNN that provides a solution for the gradient-vanishing problem in the conventional Recurrent Neural Network with gated structures.

The performance of prediction on the basis of Deep Neural Networks is further enhanced with the acquisition of more valuable spatio-temporal data. Wu et al. [15] captured the randomness and nonlinearity among traffic flow by proposing a model based on the LSTM-RNN technique. Their proposed model improves prediction accuracy as it collects better nonlinear and random traffic flow. Zhang et al. [16] proposed another model based on the Convolutional Neural Network in order to model periodic traffic data. In their proposed model, it takes the time series and folds it into a two-dimensional matrix, and this matrix is further considered as an input to CNNs. Additionally, their proposed model simulates multiscale traffic forms, and the overall situation is responsible for determining the general trend, while more comprehensive changes are determined locally. Jiang et al. [17] employed a Convolutional Neural Network along with a residual network as a deep-learning approach for traffic prediction after converting geographical data into images. Guo et al. [18], by combining the benefits of both LSTM and CNN techniques, proposed a new approach, i.e., the SRCN (spatio-temporal recurrent convolutional network), for traffic prediction and forecasting. However, a drawback of conventional Convolutional Neural Networks and Recurrent Neural Networks is their inefficiency in modelling complicated road networks. The present research can further be extended by implementing the integration approaches of Artificial Intelligence and Machine Learning [19–21].

2.2. Graph Neural Network-Based Approaches

Graph-structured information is represented by employing Neural Networks such as knowledge-based networks and social networks. For graph-structured information, the traditional Convolutional Neural Networks and Recurrent Neural Networks have also been employed. Shang et al. [22] proposed a model on the basis of CNNs which acts directly on graphs of any size and form. In contrast, Zhu and Koniusz [23] proposed a scalable method implementing a convolutional architecture on the basis of the localised estimation of spectral graph convolutions, an effective variation which can function directly on graphs. The limitation of these studies is that only undirected graphs can be used for the execution of these approaches. Graph-modelling neural networks, or GNNs, were first introduced in 2005 [24], and in recent years, their appeal and scope have increased to

a great extent. The Graph Neural Network family as a whole includes a broad range of subjects, including as reinforcement learning, unsupervised learning, supervised learning, and semi-supervised learning. For instance, a model proposed by Lu et al. [25] based on GNNs has the capability of mapping a graph G , including all its nodes, to an m -dimensional Euclidean space with $n \in N$. On the other hand, Reza et al. [26] proposed a model that is an extension of Lu et al.'s [25], in which the authors used gated recurrent elements for making output GNN sequences. Bogaerts et al. [27] proposed a GCNN model by employing CNNs for handling graph data in addition to a variety of alternative GNNs. Dong et al. [28] proposed a unified architecture based on CNNs for handling non-Euclidean structures such as three-dimensional surfaces and graphs. Yadati [29] proposed a model in which the authors combined various GNNs and GCNNs into one type of neural network and named it an MPNN (message-passing neural network). In contrast to that study, Zhan et al. [30] proposed the use of an NLNN, i.e., a non-local neural network that integrates multiple self-attention models.

Even though the idea of Graph Neural Networks has been around for a while, their advancement continues to depend on the development of the theoretical aspects and the background that comes through DNN research. The theoretical and background of RNNs serve as the foundation for creating GNNs. More specifically, in Graph Neural Networks, certain modules of the neural network are initially well-defined and later recursively processed on every edge and every node. Due to this, the executions of Graph Neural Networks typically involve more intricate operations. Therefore, the research on boosting neural-network training speeds with GPUs (Graphic Processing Units) has benefited the recent rapid development of the Graph Neural Network theory. Li et al. [31] suggested a new kind of neural network model, i.e., GN (Graph Network), which extends and summarises several kinds of GNN types, including Message Passing Neural Networks (MPNNs) and Non-local Neural Networks (NLNNs). The Graph Network generally defines unified NN units known as GN blocks, where every block is a graph–graph module that accepts a graph as an input, processes the structure while doing some computations, and then presents outputs as a graph. The blocks of the Graph Network can be arranged in a number of different ways to facilitate the creation of complex architectures. Although its usefulness has only been tested through a number of trials on simulation data rather than actual data, graph-structured knowledge has recently been used for forecasting traffic, such as in the GCNN + LSTM model for real-time traffic prediction [32]. Graph-structure modelling techniques using GNNs or GCNNs are progressing quickly; however, simulating the evolution of traffic conditions calls for a strategy in order to deal with graph sequences.

In this literature review, various approaches for predicting city-road traffic speed are discussed, including traditional statistical learning techniques and more advanced deep-learning techniques, with a specific focus on Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Additionally, the use of Graph Neural Network-based approaches for traffic prediction is explored.

Table 1 provides a tabular comparison of the different approaches mentioned in the literature review:

Table 1. Literature review comparison of the different approaches.

Approach	Description	Advantages	Limitations
Time Series [16]	Autoregressive moving-average techniques	Suitable for handling time-series data	Limited representation of complicated scenarios
Statistical Learning Techniques [12,13]	Gradient boosting, k-nearest neighbor, SVR	Aggregates relevant information	Limited improvement due to inadequate representation
Deep Neural Networks (DNNs) [14,15]	CNNs, RNNs	Better representation and performance	Inefficiency in modelling complicated road networks
LSTM-RNN [15]	Model based on LSTM-RNN for traffic flow	Captures randomness and nonlinearity	Depends on availability of spatio-temporal data
CNN-based Models [25–27]	Model based on CNNs for periodic traffic data	Simulates multiscale traffic forms	Limited to undirected graphs
GNN-based Approaches [31,32]	GNNs, GCNNs, GNs	Enables processing of graph-structured data	Complexity in operations

Traditional statistical learning techniques provide some advantages in handling time-series data, but their representation of complicated scenarios limits their performance improvement. Deep Neural Networks, such as CNNs and RNNs, have shown better representation and performance in traffic-speed prediction. However, they may not efficiently model complicated road networks. The combination of LSTM-RNN and CNN techniques has been proposed to improve traffic prediction accuracy by capturing randomness and simulating multiscale traffic forms. Graph neural network-based approaches, such as GNNs, GCNNs, and GNs, have gained attention and offer the ability to process graph-structured data. However, the limitation of handling only undirected graphs needs to be addressed. The deep-learning techniques, particularly CNNs and RNNs, show promise in improving traffic-speed prediction. The use of Graph Neural Network-based approaches provides a way to handle graph-structured information, although some limitations still need to be addressed. Integrating AI and ML approaches can be a potential direction for further enhancing traffic prediction models.

Furthermore, the authors in [33] present a Recurrent Neural Network (RNN) and satellite image-based real-time traffic-speed prediction algorithm. To increase forecast accuracy, the model integrates satellite pictures and the RNN to collect temporal trends in traffic data. The authors in [34] outline a graph convolutional network (GCN)-based traffic-flow prediction model that incorporates spatial-temporal information. Road network spatial dependencies are captured by the GCN, and historical traffic flow data are taken into account by the spatial-temporal characteristics. A real-time traffic-prediction model based on graph convolutional networks (GCNs) is presented in this paper. The model uses GCNs to represent geographical relationships in the traffic data and integrates the road network's graph structure [35]. In a study revealed in [36], a traffic-state estimation and prediction model that combines Recurrent Neural Networks (RNNs) and graph convolutional networks (GCNs) is proposed. The RNNs capture temporal correlations in the traffic data, while the GCNs collect geographical connections. A real-time traffic-speed prediction model that integrates data from several sources, such as traffic data, meteorological data, and social media data, is presented in article [37]. To identify temporal trends in the data, the model makes use of LSTM networks.

Comparing the articles in the literature review with the proposed STGGAN model, both the state-of-the-art (SOTA) models and the STGGAN model take into account temporal trends in the traffic data. While not expressly described in the SOTA model, the STGGAN model includes graph structures, edge characteristics, an RNN-based gated module, and a residual structure. The STGGAN model is also able to measure attentional relevance and capture spatio-temporal correlations. Hence, the STGGAN model stands out due to its integration of spatio-temporal data and graph structures, the inclusion of edge features, the use of an RNN-based gated module and a residual structure, and the incorporation of edge features. These components enable the STGGAN model to successfully measure attentional relevance, capture spatial dependencies, and optimise training, leading to accurate real-time traffic-flow prediction on urban road networks.

3. Preliminaries

In this work, for developing a Graph Neural Network for an urban road network, consider a weighted directed graph G . The network is made up of N nodes with M edges, and this graph shows the connections between the primary node, an adjacency matrix A , and a weight matrix W .

Several factors need to be taken into account when converting a road network into a topology graph based on its physical structure in order to provide proper representation and prevent subpar prediction accuracy. The various considerations taken in this article, together with their consequences are as follows:

A limited range of node detectors: It becomes difficult to define the link connections in places with poor detector coverage. This indicates that other data sources or approaches should be taken into account throughout the graph generation process in order to correctly

depict traffic flows and connections. The absence of node detectors in these locations can be partially made up for by integrating data from other sensor types or from sophisticated traffic monitoring systems.

Traffic direction and safety measures: Turning motions and safety measures cause traffic in different crossing lanes to flow in different directions. Investigating and taking into account the traffic direction data for each lane is needed in order to build an accurate directed graph. This indicates that in order to accurately represent actual traffic patterns, the graph needs include the precise traffic movements that are permitted or forbidden in certain lanes.

Correlations in space and pointless connections: In an urban road network, not all nearby roads exhibit robust spatial connections. Physical road network links alone may provide a large number of unnecessary connections, which would decrease forecast accuracy. The graph architecture should take into account how the traffic moves and how cars travel between various parts of the road network to handle this. It is important to consider proximity in space, where cars might likely reach nearby routes in close proximity. In order to build the highway network, researchers have also used node distances, demonstrating that the graph representation may be affected by both geographical closeness and real distance.

It is crucial to remember that static elements like distances and nearby relationships may fall short of adequately capturing the dynamic nature of traffic dynamics. The accuracy and efficiency of the graph representation can be improved by including real-time traffic data, such as traffic volumes, speeds, and congestion levels.

Problem Statement

In this article, we aim to learn a mapping function f that calculates the future traffic volumes $y(t + 1) = x_1(t + 1), x_2(t + 1), \dots, x_N(t + 1)$ given the historical traffic data of the road network with N detectors $X(t) = x_1(t), x_2(t), \dots, x_N(t)$ in the previous s time steps.

Then, we dissect the elements of the problem:

Traffic Volumes: The traffic volumes in the historical traffic statistics were gathered from N detectors at various time steps. The data for each detector is written as $X_i(t) = x_i(t_s + 1), x_i(t_s + 2), \dots, x_i(t)$, which indicates the traffic volumes seen at detector i in the preceding s time steps.

Specific Graph Structure: A graph G may be used to describe the road network, with the detectors acting as its nodes, and the edges signifying the connections or interactions between them. Different detectors' spatial relationships are captured by the particular graph structure G .

Mapping Function f : The mapping function f forecasts the future traffic volumes $y(t + 1)$ using the graph structure G and past traffic data $X(t)$ as inputs.

The objective is to learn this mapping function f in order for it to accurately estimate future traffic volumes and capture the latent spatio-temporal correlations in the historical traffic data.

In order to anticipate future traffic volumes, the traffic-flow prediction problem makes use of previous traffic data, graph structure, and a mapping function. The aim is to uncover the hidden spatio-temporal dependencies in the data to make accurate predictions.

4. Proposed Methodology

This article analyses the different traffic-speed prediction methods along with proposing a real-time traffic-speed estimation methodology that can effectively combat the challenges of the present methods reported in the literature. The proposed methods are effective in terms of improved prediction accuracy as well as minimising the model errors. The major focus of real-time traffic-speed estimation is the observation of traffic speed utilising N different sensors, and the GNN-based deep-learning model is utilised for the analysis of mass traffic data [38]. The prediction of traffic signals is performed by the graph network; therefore, the relationship between N sensors is visualised using different GNNs. While

expressing the graph of traffic speed as G_t , the graphical signal is expressed as X^t where the signal is observed over the time t . For the estimation of traffic speed, the challenge is solved by learning a function from the historic graphical signal in order to obtain the output future graphical signal. The detailed signal version is depicted in Figure 2.

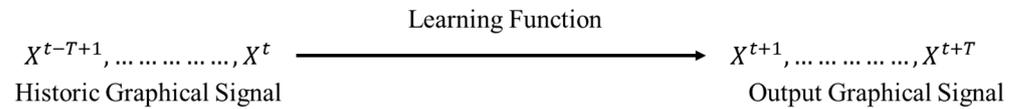


Figure 2. Traffic-speed prediction using historic and output future graphical signals.

The input X can be expressed in terms of two different feature domains. One is in terms of the spatial domain, where it is expressed as the collected features based on the basic parameters of traffic speed. However, in the temporal domain, this can be expressed as the conjunction of time-sequenced traffic-speed parameters. For defining the traffic speed, three different input parameters are considered, which are the flow of traffic, traffic density, and the speed of traffic vehicles. These are denoted as:

- Flow of Traffic: The complete number of vehicles expressed in terms of their passage from a certain place in a specific time interval. It is expressed as Equation (1).

$$\text{Flow rate} = Dv \quad (1)$$

where D is the density of traffic flow and v is the velocity of traffic vehicles.

- Traffic Density: This refers to the vehicle density at a particular date and is expressed as Equation (2).

$$\text{Density } (D) = \frac{\text{Number of vehicles}}{\text{Road length}} \quad (2)$$

- Speed of Traffic vehicles: The speed of traffic vehicles is defined as the ratio of the road segment length to the average travel time for the vehicles to pass through a road segment. The expression for the average traffic vehicle speed is expressed as Equation (3).

$$\text{Average Traffic Speed } (S) = \frac{L_{RS}}{\frac{1}{n} \sum_{i=1}^n \frac{1}{v_i}} \quad (3)$$

where L_{RS} is the length of the road section, n is the vehicle number, and v_i is the speed of travel for the i th vehicle.

4.1. Feature Analysis for Traffic-Speed Prediction

The analysis for traffic speed in this article is performed using Pearson's correlation analysis for evaluating the quantitative periodicity of the traffic speed. A heatmap is utilised to evaluate the values obtained using the Pearson correlation analysis. This analysis provides the periodic characteristic of closeness in traffic-speed data. The Pearson coefficients are utilised to measure the degree of closeness through the correlation analysis [39]. The Pearson coefficient (PC) is computing between the future predictions (F) and the historical data (H). It is expressed as Equation (4).

$$\text{Pearson Coefficient } (PC) = \frac{\sum_{i=1}^{TD} (H_i - \check{H})(F_i - \check{F})}{\sqrt{\sum_{i=1}^{TD} (H_i - \check{H})^2 (F_i - \check{F})^2}} \quad (4)$$

In this equation, TD represents the duration of the prediction, and the future predicted traffic speed is indicated by F . The historical data H is utilised for traffic-speed predictions. The mean values for both the historical and future predicted data values are represented by \check{H} and \check{F} , respectively. The value of PC ranges between -1 and 1 . A stronger positive correlation is indicated by values closer to 1 , while PC values closer to 0 indicate a

weaker correlation. The complete description of strong and weak correlations are provided in Table 2.

Table 2. Correlation Coefficients (PCs) corresponding to the degree of strong-ness.

PC Values	Degree of Correlation
$ PC \geq 0.8$	Higher degree of Correlation
$0.5 \leq PC < 0.8$	Moderate degree of Correlation
$0.3 \leq PC < 0.5$	Low degree of Correlation
$ PC < 0.3$	Not Relevant

The outcomes obtained by the Pearson correlation analysis are utilised as the objective function for confirming the intermittent constituent module. It also verifies the applicability of the universal correctness of dataset selection. Furthermore, the second component of the proposed model framework utilises the multi-layered spatial–temporal graph analysis. This analysis is performed using the multi-layered Spatio-Temporal Graph Neural Network (STGNN). This neural network module has the capability of extracting the temporal as well as spatial characteristics for traffic-speed data prediction [40]. The three main modules utilised in this article are depicted in Figure 3.

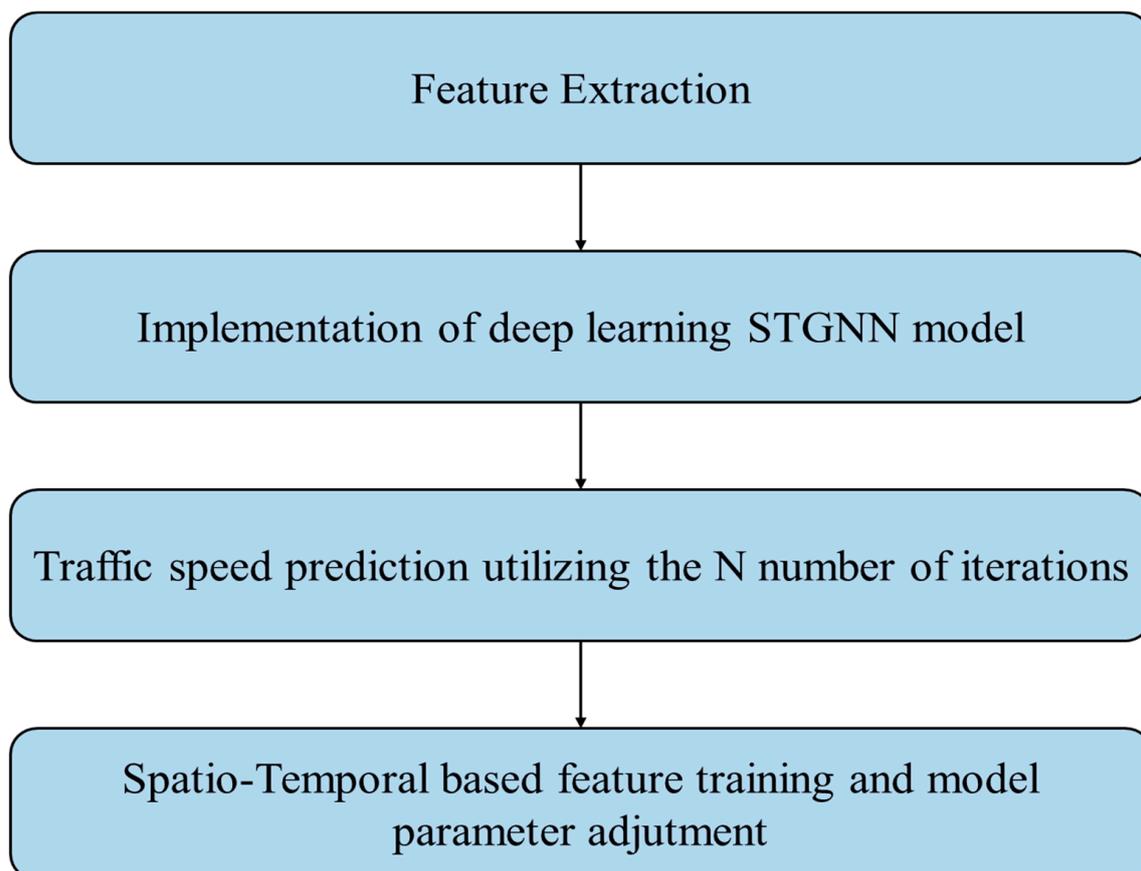


Figure 3. Main modules of the proposed STGNN framework.

The framework includes feature extraction in the first stage, followed by the aggregation of spatial features so as to provide the outcomes for the deep-learning model. The deep-learning STGNN model is utilised in the second stage of this article to obtain the extracted feature set for traffic-speed data utilisation. Furthermore, the final stage involves the traffic-speed prediction utilising the N number of iterations. The STGNN model completely learns the various spatial and temporal characteristics extracted from the features. The

significantly reliable and high-dimensional feature information is utilised for the STGNN model. Furthermore, this model utilised the spatio-temporal-based feature training, and model parameters are adjusted through the forward proceeding of the model and the back propagation of the error values. This improves the accuracy as well as reliability of the proposed traffic-speed prediction framework.

Furthermore, the key components of the STGNN-based approach consider various steps involved like the building graph representation, node and edge embeddings, message passing, and temporal dynamics involved in it, followed by prediction and evaluation.

Building Graph Representation: Building a graph representation of the road network is the initial stage in the GNN-based methodology. The connection between segments of a road is modelled as edges, and each piece of the road is represented as a node. The nodes or edges may also have other features, such as the length of the road, the posted speed limit, and past traffic information.

Node and Edge Embeddings: To capture the characteristics and connections found in the graph, GNNs need node and edge embeddings. Edge embeddings may depict the spatial relationships between linked segments, whereas node embeddings can store information about the features of individual road segments.

Message Passing: GNNs use a message-passing mechanism to collect and spread information throughout the graph. Nodes communicate with their neighbours throughout each iteration, updating their own representations in response to the messages they receive. GNNs may record both local and global dependencies inside the network thanks to this iterative approach.

Spatio-Temporal Dynamics: GNNs may be expanded to include spatio-temporal dynamics in order to perform real-time traffic-speed estimations. GNNs may learn patterns and trends over time by taking into account historical traffic data and the order of prior speed measurements, allowing precise forecasts of future traffic speeds.

Prediction and evaluation: The GNN model may be used to forecast real-time traffic speeds for various road segments in the smart city after being trained on historical data. Utilising suitable measures, such as mean squared error or mean absolute error, the model output may be assessed against ground-truth speed measurements.

The GNN-based approach for the real-time estimation of traffic speed offers improved accuracy by capturing the spatial dependencies and temporal dynamics within the road network. GNNs can provide more accurate and fine-grained predictions of traffic speed compared with traditional methods. GNNs are capable of processing data in real-time, enabling timely updates of traffic-speed estimations. This information can be used for dynamic route planning, congestion management, and for optimising traffic signal timings. These GNN-based approaches can be scaled to large road networks, making them suitable for implementation in sustainable smart cities with extensive transportation systems. The GNN framework can be modified to incorporate additional data sources such as weather conditions, event information, or sensor data from vehicles to enhance the accuracy of traffic-speed estimation. These applications of GNN-based approaches for real-time traffic-speed estimation aligns with the goals of sustainable smart cities, promoting efficient transportation, reducing congestion, and improving overall urban mobility. By leveraging the power of deep learning and graph analysis, these approaches contribute to building smarter, greener, and more sustainable cities.

4.2. Methodological Framework

The previous section provided an overview of the traffic-speed prediction analysis and identified the research gaps in the recent literature. The objective is to estimate the traffic speed in real-time using a GNN-based approach. This involves predicting the speed of vehicles at different locations in the city based on historical and current data. For this purpose, data collection is performed including the historical traffic-speed records, sensor data (e.g., from traffic cameras or GPS devices), road network information (e.g., road segments, intersections), and any additional contextual data (e.g., weather

conditions, time of day). After data collection, handle any missing values and perform any necessary transformations or feature engineering. Convert the data into a suitable format for GNN-based models. Then, represent the network as a Spatio-Temporal Gated Graph Attention Network (STGGAN); here, the edges between the nodes define their connectedness based on the structure of the road network, while the nodes themselves represent road segments or intersections.

The STGGAN model contains a Gated Recurrent Unit (GRU) layer for capturing temporal correlations, a GAT layer with edge features for modelling spatial dependencies, an RNN-based gated module for assessing attention and significance, and a residual structure for increased training effectiveness. Combining both spatio-temporal data and graph structures, these elements allow the model to accurately estimate short-term traffic flow on urban road networks. For the purpose of predicting traffic flow on metropolitan road networks, the STGGAN (Spatio-Temporal Graph Attention Network) model is presented. To identify spatio-temporal relationships and enhance the model's functionality, it includes a number of components. Using a GRU (Gated Recurrent Unit) layer, temporal correlations are captured. The model can learn temporal patterns in the data thanks to the GRU layer's assistance in capturing dynamic variation features across past traffic conditions. The self-attention mechanism in the model adds edge characteristics for modelling spatial interactions. This feature increases the model's capacity to capture spatial relationships between various sections by supplying prior knowledge about the characteristics of the road network. In order to assess the significance of various elements inside the multi-head attention mechanism, an RNN-based gated module is presented. During the prediction phase, this method enables the model to concentrate on pertinent information and to distribute attention efficiently. The STGGAN model adopts the residual readout layer structure, inspired by residual mapping, to hasten convergence and to capture minute changes. A residual structure increases the forward speed of information by allowing it to transcend many layers, allowing the model to efficiently optimise and capture the identity mapping.

Thus, the proposed STGGAN model combines a GRU layer, edge features, an RNN-based gated module, and a residual structure in contrast to the naïve GAT model. By collecting both the geographical and temporal linkages in traffic data, this improves the model's capacity to derive spatio-temporal dependencies. The STGGAN model's capacity to simulate the connections between various parts of the road network is improved by using edge features for a directed arc to better capture spatial interactions. A directed arc in a transportation network denotes a link or connection between two points or nodes. The model may include more details about the properties of the road segment that is represented by a directed arc by taking edge features into account. These edge features may comprise characteristics of the road segment, such as the kind of road, the speed limit, the amount of traffic, the state of the road, or any other pertinent data. The model may benefit from past knowledge of the road network's properties by including edge features for a directed arc. The model may better comprehend the spatial interconnections and interactions between various road segments by taking these properties into account. When a model discovers, for instance, that some road types or situations often have a greater influence on traffic flow than others, it may adjust its forecasts appropriately. By merging the spatial and temporal components of the road network, edge characteristics in the STGGAN model result in a more thorough description of the road network. The GRU layer's dynamic changes and the edge features' spatial interactions are both taken into account in the model in order to properly predict short-term traffic flow on urban road networks in real-time. Therefore, using edge features for a directed arc in the STGGAN model facilitates the incorporation of spatial data and improves the model's capacity to capture the intricate relationships between various parts of the road network, resulting in more precise traffic-flow predictions in sustainable smart cities.

For defining the Spatio-Temporal Gated Graph Attention Network (STGGAN) architecture, use the following steps.

- Define the node features: assign initial features to each node, such as historical traffic-speed values or contextual information.
- Define the edge features: compute edge features based on the relationship between nodes, such as distance or road connectivity.
- Initialise node and edge representations: assign initial representations to nodes and edges.
- Gating Mechanism: Introduce a gating mechanism to control the flow of information in the network. This can be achieved by using Gated Recurrent Units (GRUs) or other gating mechanisms.
- Attention Mechanism: Implement a graph attention mechanism to capture the rank of neighbouring nodes for each node's representation. This can be performed using attention coefficients that determine the relevance of neighbouring nodes during message passing.

Furthermore, model training is performed, which involves splitting data into training, validation, and testing sets. Initialise the model and then train it by using the training data to optimise it to minimise the prediction error. Furthermore, tune hyperparameters and monitor the model's performance on the validation set to prevent overfitting. For real-time estimation, the trained network model is continuously updated for the graph representation and by feeding the new data into the STGGAN model to make predictions for traffic speed at different locations and updating the predictions as new data becomes available. Also, the model's performance is assessed using appropriate evaluation metrics, such as mean absolute error or root mean square error. Compare the proposed approach with baseline methods or other relevant approaches. For the integration of the STGGAN-based traffic-speed estimation model with other smart city systems, such as traffic management systems or intelligent transportation systems, enable real-time monitoring and decision making based on the estimated traffic speeds. The architecture of the proposed STGGAN-based network designed for this research work is depicted in Figure 4.

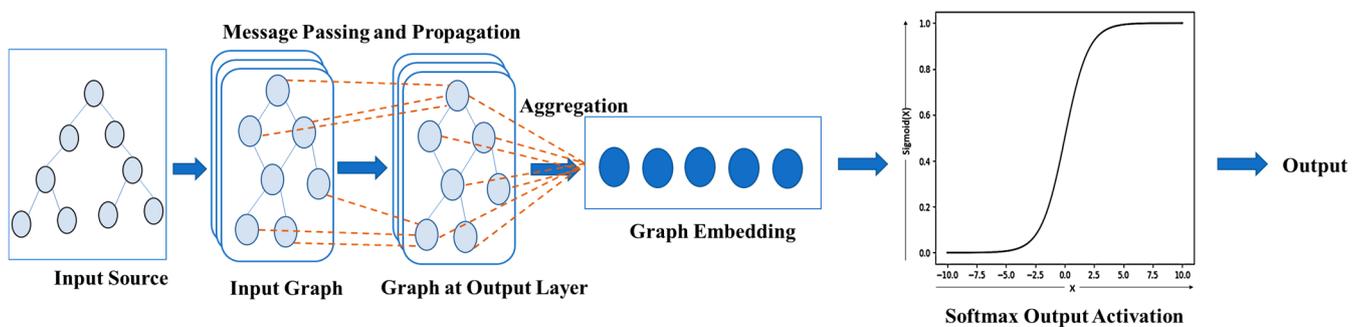


Figure 4. Architecture of proposed STGGAN-based network.

The proposed framework depicts the traffic speed for overcoming the shortcomings of the existing models. The interoperability of the input data is also improved along with the improvement in the learning ability of the deep-learned model for higher-order correlation analysis [41]. Also, the proposed model is effectual in extracting the dynamic association between the traffic structure networks and the nodes. The proposed model framework is depicted in Figure 5.

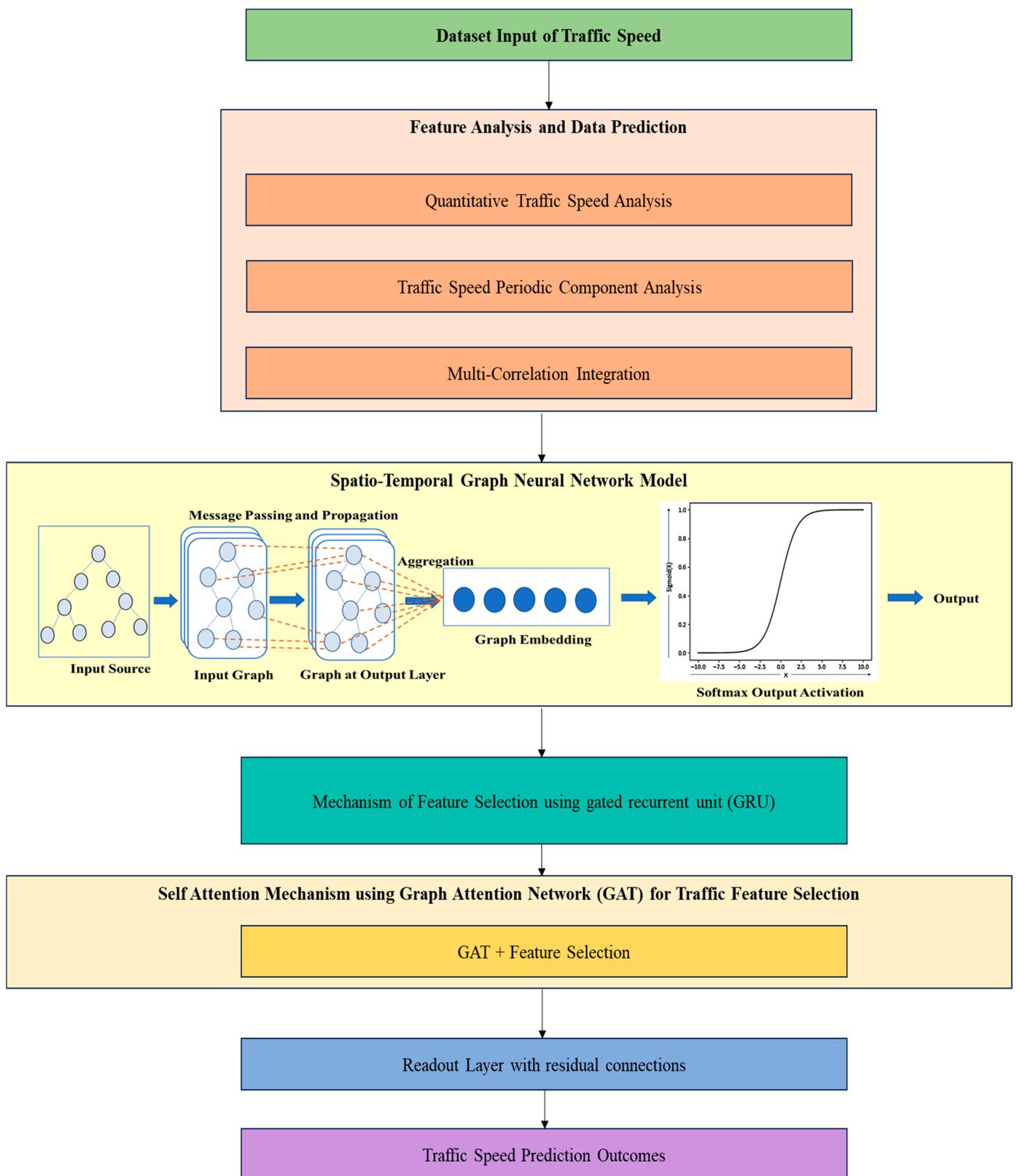


Figure 5. Model framework for the proposed STGGAN method.

The proposed model comprises three different modules. The initial component is responsible for feature analysis as well as the periodic prediction of the data after fusion. The correlation analysis is performed for outputting the characteristic data determined

after the feature analysis. Furthermore, the second component of the proposed framework is a convolution analysis based on the spatial–temporal Graph Neural Network. For the convolution analysis, this work utilises a multi-layered Graph Neural Network model so as to obtain a higher dimensionality in terms of the spatio-temporal feature processing obtained from the first component stage [42]. The feature-selection-stage mechanism is introduced in the third component of the proposed model in order to improve the feature reliability of the extracted set of features for traffic-speed data.

4.3. Spatial Dependency

In order to anticipate traffic flow on urban road networks, knowledge of spatial relationships is essential. The Graph Attention Network (GAT) is used in this portion to capture these spatial relationships. The GAT has shown success in a variety of tasks, including recommendation systems and computer vision. The GAT offers a number of benefits over the popular Graph Neural Network (GNN) model for predicting traffic flow:

1. **Weight Allocation:** The GAT can handle directed graphs and may give different weights to various neighbours, which is useful for metropolitan road networks with direction-specific traffic flows.
2. **Node characteristics:** Rather than the graph structure itself, model parameters in GAT are tied to each node's characteristics. The GAT may be used for inductive learning tasks thanks to this trait, which allows the model created for one set of road networks to be used to forecast traffic conditions on other road networks.

For the purpose of defining the GAT, we refer to the input characteristics of node i at layer l as h_i^l , where h_i^l denotes the input's prior traffic states, or X_t^i . The following is the GAT formulation indicated in Equations (5)–(7):

$$y_i^l = W^l \times h_i^l \quad (5)$$

where W^l is the learnable weight matrix;

$$e_{ij}^l = \text{Leaky ReLU}\left(a^T \left[y_i^l \parallel y_j^l \right]\right) \quad (6)$$

where a^T is a single-layer feed-forward network; and

$$\text{softmax}\left(e_{ij}^l\right) = \frac{\exp\left(e_{ij}^l\right)}{\sum \exp\left(e_{ij}^l\right)} \quad (7)$$

The urban road network is transformed into a weighted directed graph for the road-network transformation method. Edge characteristics are disregarded in the aforementioned GAT layer as previous information to enhance the computation of attention weights in order to take use of the edge features. As a result, to optimise the process of computing the attention weights, we add the edge weights E of the created weighted directed graph as prior knowledge. Additionally, including edge features into the GAT layer helps to lessen the effects of unused connections brought on by mistakes during construction. Thus, the softmax function is indication in Equation (8).

$$\text{softmax}\left(e_{ij}^l\right) = \frac{\exp\left(E_{ij} \times e_{ij}^l\right)}{\sum \exp\left(E_{ij} \times e_{ij}^l\right)} \quad (8)$$

where E_{ij} is the edge weight.

4.4. Temporal Dependency

Given that they represent the time-varying properties of network flow, temporal dependencies are essential for effectively forecasting future traffic states. Instead of relying only on a straightforward linear transformation in this work, we suggest using the Gated Recurrent Unit (GRU) to properly capture these temporal connections.

A linear transformation is employed to convert the lower-level characteristics into a high-dimensional space. However, as the input characteristics are essentially time-series data reflecting prior traffic conditions, this technique falls short of thoroughly examining the temporal evolution of the input features. We make use of GRU, a popular Recurrent Neural Network (RNN) variation recognised for its capacity to represent sequential input, to overcome this constraint.

Two essential parts make up a GRU block: a reset gate (r_t) and an update gate (u_t). The update gate aids in capturing long-term dependencies in the time series, while the reset gate is in charge of eliminating old data that is unrelated to future states. The GRU layer may be stated as Equations (9) and (10) for an input feature i_t :

$$r_t = \sigma(W_r i_t + W_r H_{t-1}) \quad (9)$$

where W is the weight matrix for reset gate and $\sigma(x) = \frac{1}{1+\exp(-x)}$; and

$$u_t = \sigma(W_u i_t + W_u H_{t-1}) \quad (10)$$

where W is the weight matrix for update gate and $\sigma(x) = \frac{1}{1+\exp(-x)}$.

4.5. Spatio-Temporal Prediction Model Framework

For the purpose of predicting short-term traffic flow on urban road networks, the STGGAN (Spatio-Temporal Graph Attention Network) model is the output. A feature analysis and data prediction layer, a Gated Recurrent Unit (GRU) layer, a feature selection layer, and a GAT (Graph Attention Network)-based self-attention mechanism make up its four primary parts. The model seeks to capture both the geographical relationships between various road network segments and the temporal correlations in historical traffic data.

The feature analysis and data prediction layer, the initial element, is used to enhance the model's capability of detecting temporal correlations. A Gated Recurrent Unit (GRU) layer aids in identifying sequential patterns in the data. Next, the GAT (Graph Attention Network)-based Graph Neural Network (GNN) model is enhanced using edge characteristics. The GAT is a Graph Neural Network design that can simulate the connections between various graph nodes. The Spatio-Temporal STGGAN model improves the spatial interdependence between various road network segments by including edge features, taking into account both traffic levels and the properties of the individual road segments. The value of various heads in the multi-head attention mechanism is assessed using the Recurrent Neural Network (RNN)-based gated module. When producing predictions, the model's attention processes enable it to concentrate on pertinent portions of the data. Based on the input data, the gated module assists in dynamically allocating priority to various attention heads, enhancing the model's capacity to pay attention to pertinent information.

Additionally, a readout layer with residual connections is adopted by the STGGAN model. Residual connections speed up training and increase convergence efficiency by allowing the model to learn from discrepancies between anticipated and actual values. This structure aids the model's efficient learning and gradient propagation during training.

The algorithm for the Spatio-Temporal Graph Neural Network (STGNN) model uses a Gated Graph Attention Network-based STGGAN approach for the real-time estimation of traffic speed in sustainable smart cities. The goal is to predict the speed of vehicles at different locations in the city using historical and current data, taking into account the graph structure of the road network. The algorithm for GNN modelling is indicated as Algorithm 1:

Algorithm 1: Spatio-Temporal Gated Graph Attention Network (STGGAN)**Input:**

X: Node features ($\text{node_count} \times \text{feature_dim}$)
 A: Edge features ($\text{node_count} \times \text{node_count} \times \text{feature_dim}$)

Parameters:

K: Number of attention heads
 hidden_dim: Hidden dimension
 output_dim: Output dimension
 Initialise weight matrices: W_{node} ($\text{feature_dim} \times \text{hidden_dim}$), W_{edge} ($\text{feature_dim} \times \text{hidden_dim}$),
 W_{att} ($\text{hidden_dim} \times \text{hidden_dim}$) for each attention head, and W_{out} ($K \times \text{hidden_dim} \times \text{output_dim}$)

Initialisation:

$H = \text{zeros}(\text{node_count}, \text{hidden_dim})$ # Node representations
 $E = \text{zeros}(\text{node_count}, \text{node_count}, \text{hidden_dim})$ # Edge representations

Node representation computation:

for k in range(K): # Attention heads
 for i in range(node_count): # Nodes
 $h = \text{GatingMechanism}(H[i], X[i])$
 $H[i] = h$

Edge representation computation:

for k in range(K): # Attention heads
 for i in range(node_count): # Source nodes
 for j in range(node_count): # Destination nodes
 $e = \text{LeakyReLU}(H[i] * W_{\text{edge}} + H[j] * W_{\text{edge}})$

$E[i, j] = e$

Attention mechanism:

for k in range(K): # Attention heads
 $H = \text{AttentionMechanism}(H, E)$

Output computation:

$\text{output} = \text{softmax}(H * W_{\text{out}})$

return output

Function GatingMechanism(h, x):

Gated Recurrent Unit (GRU) for gating mechanism

$r = \text{sigmoid}(W_r * x + U_r * h)$

$z = \text{sigmoid}(W_z * x + U_z * h)$

$h_{\text{hat}} = \text{tanh}(W_h * x + U_h * (r * h))$

$h_{\text{new}} = (1 - z) * h + z * h_{\text{hat}}$

return h_{new}

Function AttentionMechanism(h, adj):

Graph attention mechanism

$e = \text{LeakyReLU}(\text{Attention}(h * W_{\text{att}}, h * W_{\text{att}}))$

$\alpha = \text{softmax}(e)$

$h_{\text{new}} = \text{sum}(\alpha * h)$

return h_{new}

Here, the activation functions (e.g., sigmoid, tanh, LeakyReLU, softmax) and the weight matrices (e.g., W_{node} , W_{edge} , W_{att} , W_{out}) are assumed to be defined and initialised appropriately. The GatingMechanism and AttentionMechanism functions are defined separately to capture the specific computations involved in those steps.

This work provides an effective analysis of the data with traditional models. The proposed model is better compared with the traditional models in terms of data quality as well as input reliability. The system scalability is also improved in this proposed model as the calculation process is simplified and the connections are enhanced between different components. Thus, the proposed model is improved in terms of interpretability and accuracy.

5. Results and Analysis

The experimental analysis of the proposed STGNN-based STGGAN approach for real-time traffic-speed estimation while implementing it for smart cities was performed in this article using two different actual traffic datasets. The datasets utilised in this article are the PeMSD4 [43,44] and PeMSD8 [43,44] datasets, which comprise a varying number of samples and are sampled using a different number of sensors. The PeMSD4 dataset was acquired using 307 sensors which resulted in a total of 16,992 sample outcomes; however, the PeMSD8 dataset was acquired using 170 sensors and resulted in 17,856 sample sets. Feature details of the two datasets involved in this study are depicted in Table 3.

Table 3. Feature detailing of the PeMSD4 and PeMSD8 datasets involved in this study.

PeMSD4 Dataset [43,44]	PeMSD8 Dataset [43,44]
Number of Sensors: 307 Total Samples: 16,992	Number of Sensors: 170 Total Samples: 17,856
Features: The dataset likely includes various traffic-flow measurements, such as traffic speed, traffic volume, and occupancy, captured by the sensors at different locations within the road network. Additionally, it may include temporal information to account for the time aspect of traffic patterns.	Features: The dataset is expected to contain similar traffic-flow measurements as the PeMSD4 dataset, including traffic speed, volume, and occupancy. Temporal information might also be present to capture time-related patterns in traffic behaviour.

A measure of traffic flow and traffic speed for a density of approx. 500 vehicles per kilometre is depicted in Figure 6. Based on the traffic flow and traffic speed indicated in Figure 6, further measures were computed in terms of prediction accuracy (PA), mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

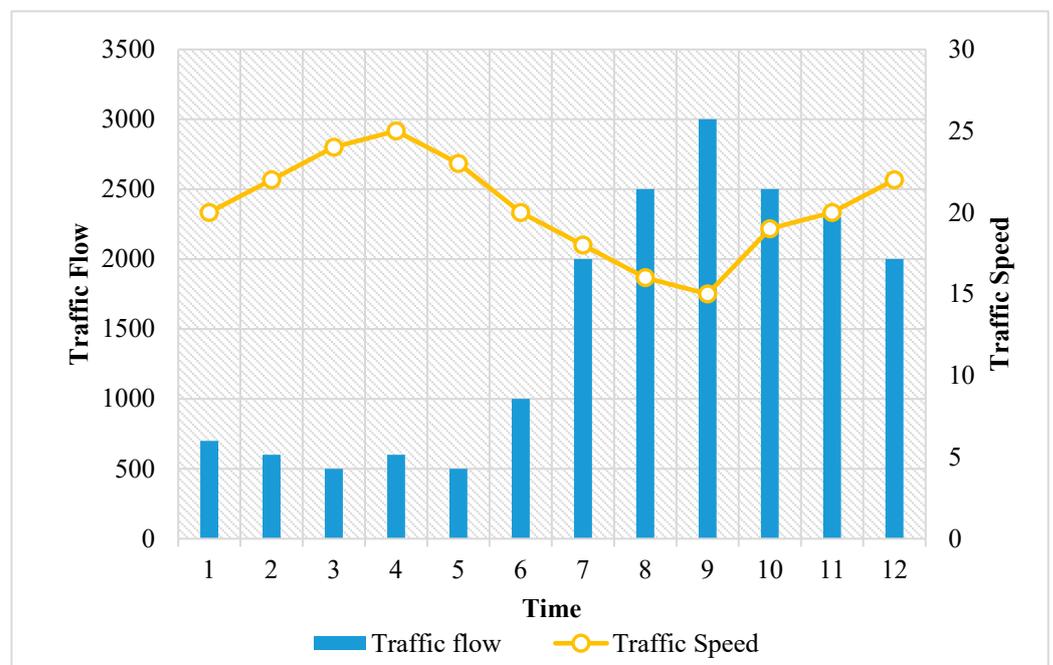


Figure 6. Traffic flow and traffic speed.

For accurate prediction using the proposed STGNN-based STGGAN approach, the network performance was tested for its prediction accuracy which was computed in terms of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values obtained for both the PeMSD4 and PeMSD8 datasets. TP refers to the situation when

the STGNN correctly predicts a positive outcome. In the case of traffic-speed estimation, it means that the STGNN accurately predicts a high traffic speed when the actual speed is indeed high. It represents the number of correctly identified high-traffic-speed instances. TN occurs when the GNN correctly predicts a negative outcome. In traffic-speed estimation, it means that the STGNN accurately predicts a low traffic speed when the actual speed is indeed low. TN represents the number of correctly identified low-traffic-speed instances. An FP happens when the STGNN incorrectly predicts a positive outcome. In the context of traffic-speed estimation, it means that the STGNN predicts a high traffic speed when the actual speed is low. FP represents the number of instances where the STGNN wrongly identifies high traffic speed. FN occurs when the STGNN incorrectly predicts a negative outcome. In the case of traffic-speed estimation, it means that the GNN predicts a low traffic speed when the actual speed is high. FN represents the number of instances where the GNN fails to identify high traffic speed. These definitions are based on the concept of binary classification, where the task is to categorise instances into two classes: high traffic speed and low traffic speed. The GNN is trained to predict the traffic speed based on the input graph data, and TP, TN, FP, and FN are computed by comparing the GNN's predictions with the actual traffic-speed labels. By analysing the values of TP, TN, FP, and FN, various evaluation metrics can be computed to assess the performance of the GNN model, and these metrics provide insights into the model's ability to correctly identify high and low traffic speeds, which are crucial for real-time traffic-speed estimation in sustainable smart cities.

This section presents a comparative analysis of the proposed STGNN-based STGGAN approach with some standard baseline models in order to assess the system's performance. Furthermore, a vast comparative study was conducted using various statistical, analytical, machine-learning- and deep-learning-based approaches. The different evaluation criteria used in this article are PA, MSE, RMSE, MAE and MAPE.

The formulas for all the evaluation indicators are provided in Equations (11)–(15).

$$PA = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n |\hat{x} - x_i|^2 \quad (12)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |\hat{x} - x_i|^2} \quad (13)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{x} - x_i| \quad (14)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \frac{|\hat{x} - x_i|}{x_i} \quad (15)$$

where x_i is the true value obtained, \hat{x} is the predicted value of the sample outcome, and n is the total sample number.

Furthermore, in this section, the impact of traffic speed at varying spatio-temporal locations is demonstrated and discussed using the model prediction accuracy for both PeMSD4 and PeMSD8 datasets. The computed prediction accuracy is shown in Figure 7.

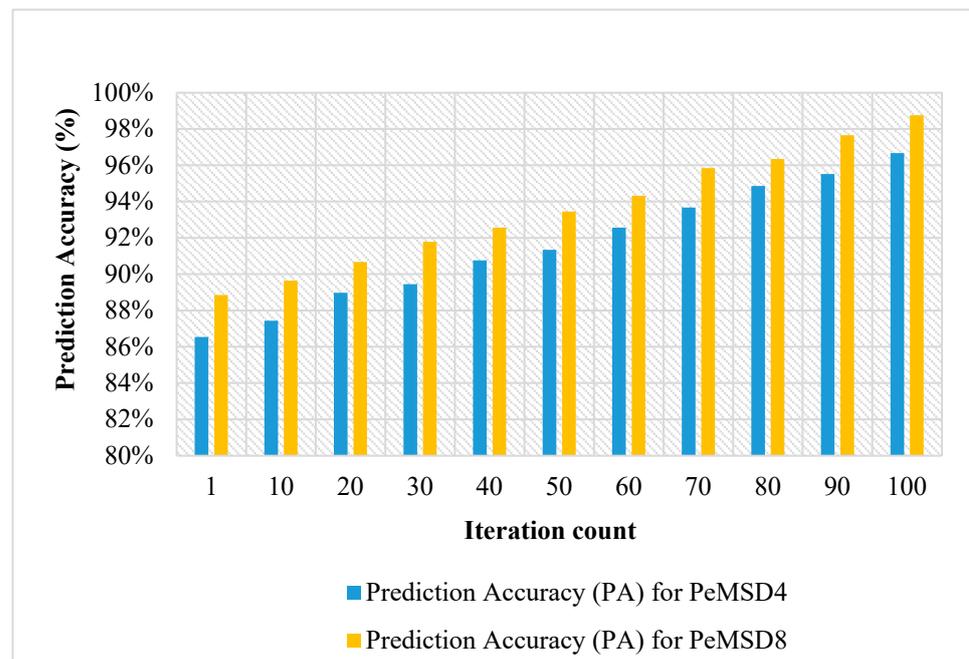
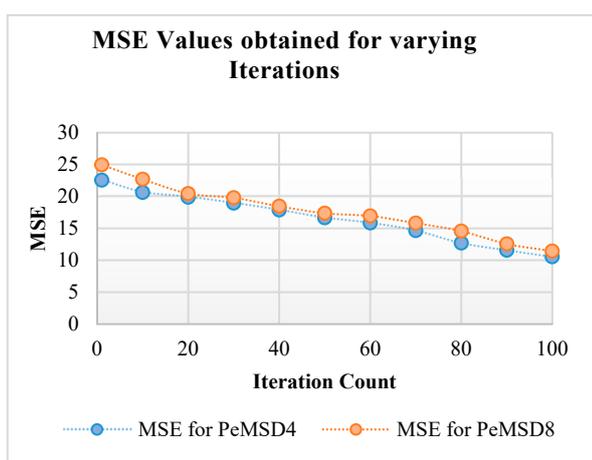
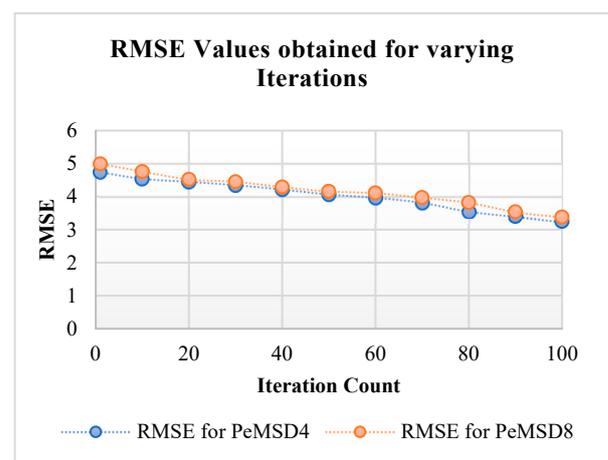


Figure 7. Prediction accuracy obtained for the proposed STGGAN model using the PeMSD4 and PEMS8 datasets.

Figure 7 shows that the proposed STGGAN model is capable of improving the accuracy of traffic-speed predictions. As the number of iterations increased from 1 to 100, the accuracy of predictions for both the PeMSD4 and PEMS8 datasets improved steadily. The accuracy values for PeMSD4 increased from 86.54% to 96.67%, and those for PEMS8 increased from 88.85% to 98.75%. The proposed STGGAN model demonstrates accurate prediction abilities for light to moderate traffic flow. Consistently high prediction accuracy for these traffic volumes indicates the model's ability to accurately predict traffic speed under such conditions. Furthermore, the error computation for the proposed STGGAN method was performed in terms of MSE and RMSE, which is revealed in Figure 8a,b.



(a)



(b)

Figure 8. MSE values (a) and RMSE values (b) observed for the PeMSD4 and PEMS8 datasets.

The MSE values observed from Figure 8a shows that the error value reduces with an increasing iteration count, and it is still decreasing at a significant rate at the 100th iteration count. For the PEMS4 dataset, an MSE value of 22.54 was observed for the 1st iteration, and this continued decreasing until an MSE value of 10.54 was obtained for the 100th

iteration count. Similarly, for the PEMS8 dataset, an MSE value of 24.95 was observed for the 1st iteration, which continued reducing until an MSE value of 11.43 was obtained for the 100th iteration count. The observations made from Figure 8b reveal that the RMSE error values gradually reduce with the increase in the iteration count from 1 to 100. An RMSE value of 4.54 was observed for the 1st iteration which continued decreasing until reaching an RMSE value of 3.25 for the 100th iteration count for the PEMS4 dataset. Similarly, for the PEMS8 dataset, an RMSE value of 4.99 was observed for the 1st iteration which continued decreasing until an MSE value of 3.38 was obtained for the 100th iteration count. As the iteration count increases from 1 to 100, the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) values decrease progressively. With increased iterations, the proposed STGGAN model obtains greater accuracy in predicting traffic speed, as indicated by this decrease.

Figure 9a,b depict the MAE and MAPE values observed for the PeMSD4 and PEMS8 datasets, respectively. In Figure 9a, MAE values are observed from the 1st to the 100th iteration count. For the PEMS4 dataset, an MAE value of 30.45 was observed for the 1st iteration which reduced to an MAE value of 20.45 for the 100th iteration count. Similarly, for the PEMS8 dataset, an MAE value of 32.48 was observed for the 1st iteration which continued decreasing to 22.34 for the 100th iteration count. The observed computation reveals that the MAE value decreases with the increasing iteration count, and it decreases to a significantly much lower value at the 100th iteration. Figure 9b depicts the MAPE values for the PeMSD4 and PEMS8 datasets. Utilising the PEMS4 datasets, an MAPE value of 22.56 was observed for the 1st iteration, which minimised to 12.45 for the 100th iteration. For the PEMS8 dataset, an MAPE value of 24.69 was observed for the 1st iteration while being minimised to 14.32 for the 100th iteration count. This reveals that MAPE error values are gradually reduced with the increasing iteration count. There are certain baseline models which are analysed in this work for the validation analysis of the proposed STGGAN approach. Several baseline models, including Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), Artificial Neural Network (ANN), Traditional Convolutional Neural Network (CNN), and Multi-Layered Graph Neural Network (STGNN), are contrasted with the proposed STGGAN model. The comparison consists of prediction error, execution time, and evaluation metrics, including MSE, RMSE, MAE, and MAPE.

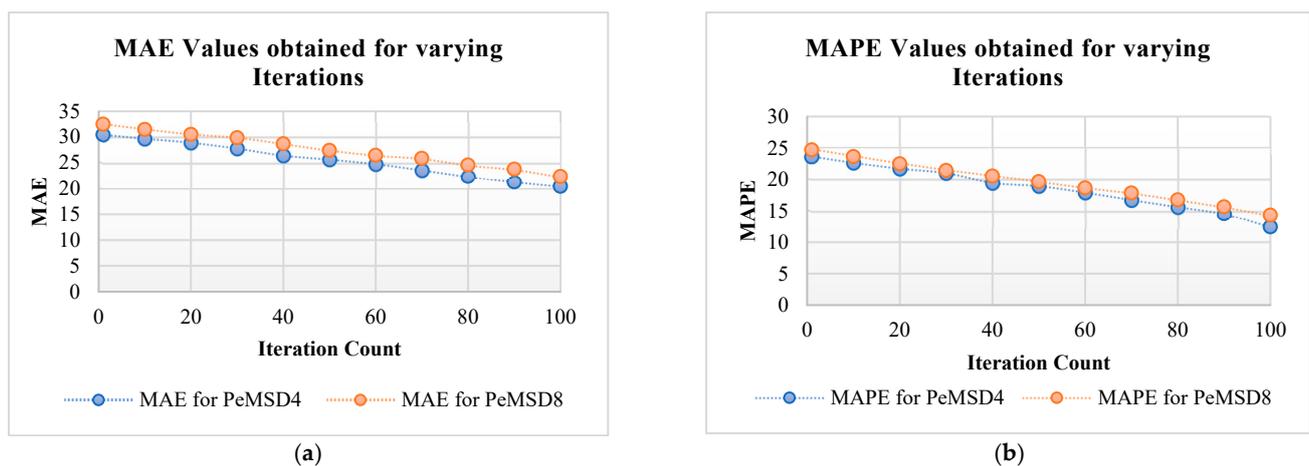


Figure 9. MAE values (a) and MAPE values (b) observed for the PeMSD4 and PEMS8 datasets.

The comparison of prediction error and execution time is provided in Figure 10 and of MSE, RMSE, MAE and MAPE errors in Figure 11.

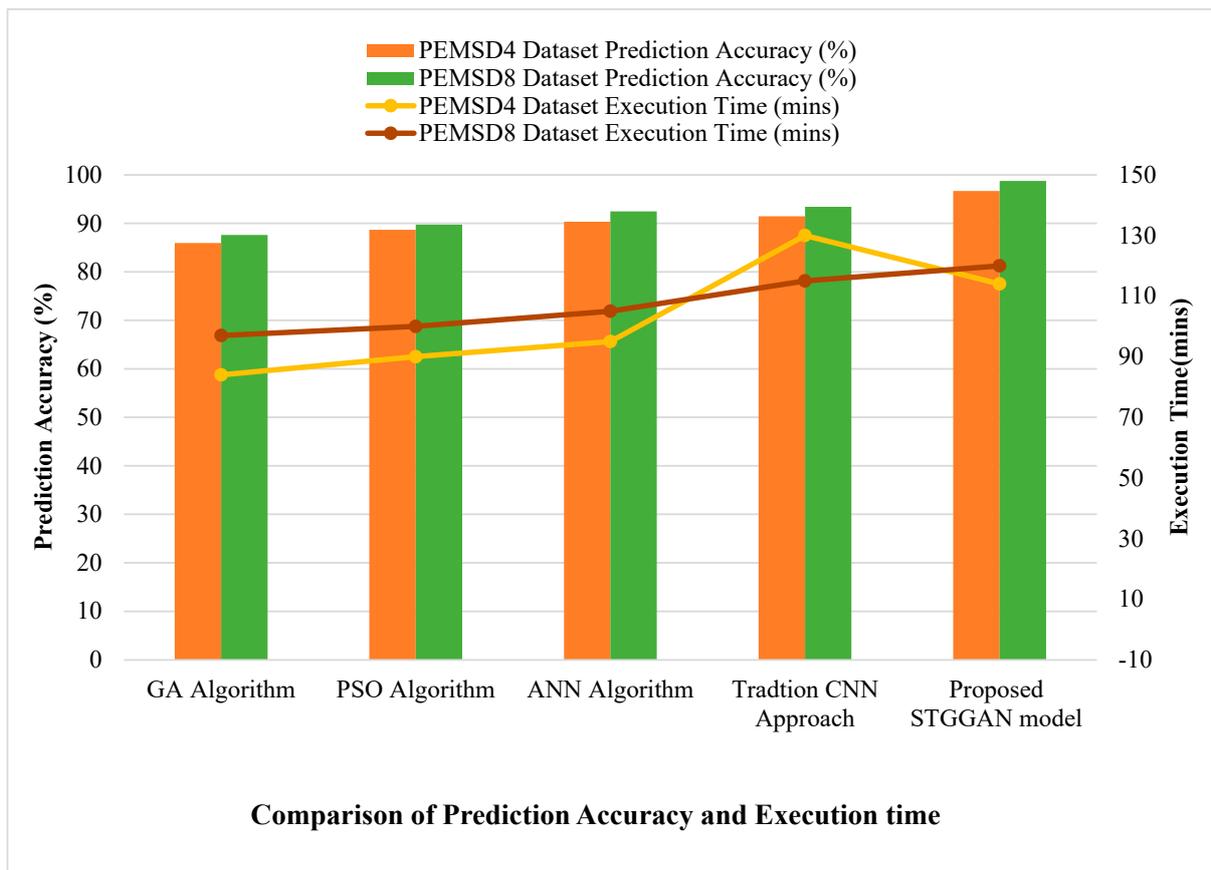


Figure 10. Comparative analysis for the proposed STGGAN model with state-of-the-art approaches [12–14,25–27] in terms of prediction accuracy and execution time.

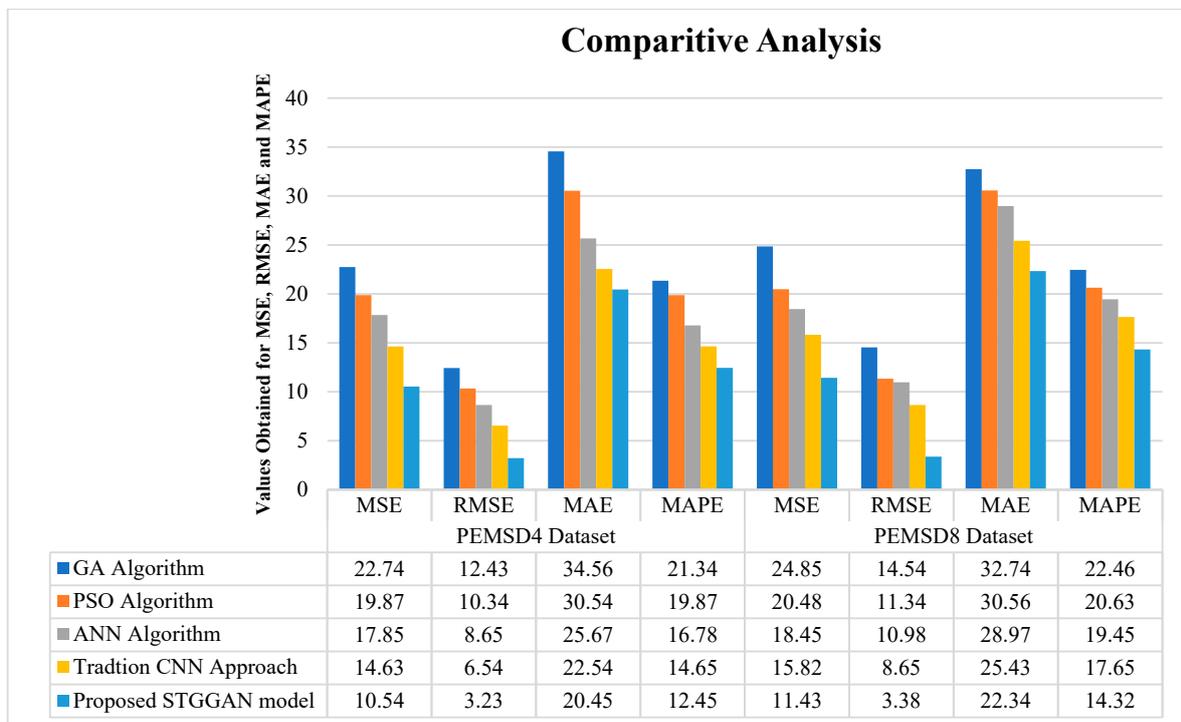


Figure 11. Comparative analysis of state-of-the-art methods [12–14,25–27] with the proposed STGGAN model.

The comparative analysis demonstrates that the proposed STGGAN model obtains greater prediction accuracy than other algorithms for both the PeMSD4 and PeMSD8 datasets. For PeMSD4, the proposed STGGAN model achieves a prediction accuracy of 96.67%, outperforming the GA algorithm (85.94%), the PSO algorithm (88.67%), the ANN algorithm (90.34%), and the conventional CNN approach (91.45%). Similarly, for PeMSD8, the proposed STGGAN model achieves a prediction accuracy of 98.75%, outperforming the GA algorithm (87.59%), the PSO algorithm (89.74%), the ANN algorithm (92.45%), and the conventional CNN approach (93.45%).

According to the execution time analysis, the proposed STGGAN method required 114 min for the PeMSD4 dataset and 120 min for the PeMSD8 dataset. These values indicate the computational efficacy of the proposed STGGAN model, even though the execution time is not explicitly compared with other algorithms.

The comparative analysis of the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) for both datasets demonstrates that the proposed STGGAN model achieves better values than the other algorithms. Lower MSE, RMSE, MAE, and MAPE values indicate greater precision and accuracy in traffic-speed prediction. The effectiveness of the proposed STGGAN model is further supported by the fact that all these metrics exhibited significant enhancements. The proposed STGGAN method exploits the spatial and temporal dependencies in the data by utilising the characteristics of a Graph Neural Network. This incorporation enables the model to capture and exploit the intricate relationships and patterns present in traffic data, which contributes to its enhanced performance. The technical observations emphasise the practicability and viability of the proposed STGGAN method. Compared with other algorithms, the model yields more accurate predictions and enhanced evaluation metrics, demonstrating its reliability. These results validate the efficacy and applicability of the proposed STGGAN model for predicting traffic speed.

In terms of prediction accuracy, evaluation metrics, and usability, the proposed STGGAN model outperforms other algorithms, according to the technical observations. Its enhanced performance is a result of the incorporation of Graph Neural Network characteristics, making it a promising approach for traffic-speed prediction tasks.

The Spatio-Temporal Traffic Prediction Model (STGGAN) framework has a number of significant advantages over other alternative methods for traffic prediction utilising Graph Neural Networks (GNN) reported in the literature [25–27]. A few of the significant advantages are listed below:

1. **Comprehensive Integration:** The STGGAN framework incorporates both spatio-temporal data and graph structures, enabling an all-encompassing approach to traffic prediction. This comprehensive integration enables the model to encompass the complex relationships between spatial and temporal factors, resulting in more accurate forecasts.
2. **Capturing Temporal Correlations:** The STGGAN framework includes a Gated Recurrent Unit (GRU) layer that effectively captures temporal correlations in the traffic data. By taking into account dynamic variations across prior traffic conditions, the model can learn and adapt to temporal patterns, thereby improving its accuracy of prediction.
3. **Modelling Spatial Dependencies:** With the incorporation of a Graph Attention Network (GAT) layer with edge characteristics, the STGGAN framework excels at modelling spatial dependencies. This feature allows the model to utilise prior knowledge about the road network, such as road types, speed limits, and other characteristics, in order to capture the complex interactions between various road sections.
4. **Attention Evaluation:** The STGGAN framework employs an RNN-based gated module to evaluate the attention and significance of various elements within the multi-head attention mechanism. This mechanism enables the model to focus on pertinent information and effectively distribute attention, thereby enhancing its overall prediction performance.

5. **Residual Structure:** Motivated by residual mapping, the readout layer of the STGGAN framework employs a residual structure. This structure enables faster convergence and the capture of minute changes in the traffic data, thereby enhancing the model's optimisation capabilities and allowing identity mapping to be effectively captured.
6. **Real-Time Estimation:** The STGGAN framework's ability to estimate short-term traffic flow in real time is one of its primary strengths. By integrating spatio-temporal data and graph structures, the model is able to provide accurate predictions in a timely manner, thereby facilitating smart city traffic management and decision making.

Thus, by integrating geographical and temporal data, using previous data, adjusting to real-time changes, and offering scalability and interpretability, the Spatio-Temporal Traffic Prediction Model framework offers a holistic approach to traffic prediction. It is a potential option for precise and effective traffic forecasting using GNNs because of these benefits. These benefits distinguish the STGGAN framework from alternative approaches and demonstrate its efficacy in addressing the challenges of traffic prediction. The comprehensive integration of spatio-temporal data and graph structures, as well as the incorporation of temporal correlations, spatial dependencies, attention assessment, and residual structure, all contribute to the superior real-time traffic flow estimation performance of the STGGAN model.

6. Conclusions

In this article, a model for predicting traffic speed based on Graph Neural Networks (GNNs) is put forth. Inspired by GNNs, a model is presented that feeds the full road topology graph into a neural network that can be tuned to predict traffic speeds. The proposed model was designed by considering the characteristics and organisation of a road network. By utilising GGAN blocks, we enhanced the performance of traditional GNN algorithms by making sequential graphs through the conversion of input and output vectors. The performance of the proposed model was evaluated through experimental analysis and compared with existing state-of-art models. It was observed through experimentation that our proposed model decreases the training loss more quickly. The extensive experimental tests were conducted using large-scale, real-world datasets. In this article, the proposed STGGAN-based approach for real-time traffic-speed estimation was implemented using two different actual traffic datasets, PeMSD4 and PeMSD8. For the PeMSD4 and PeMSD8 datasets, prediction accuracy values of 96.67% and 98.75%, respectively, were observed. It was also observed from the experimental analysis utilising both datasets that the MSE, RMSE, MAE, and MAPE errors gradually reduces with the increasing iteration count. A comparative analysis of the proposed STGGAN methodology was also performed employing various statistical, analytical, and machine-learning- and deep-learning-based approaches, which revealed the viability, reliability, and practicability of the proposed approach.

Author Contributions: Conceptualization, A.S. (Amit Sharma) and A.S. (Ashutosh Sharma); methodology, P.N. and V.G.; software, A.S. (Amit Sharma), P.N. and V.G.; validation A.S. (Amit Sharma), P.N. and V.G.; formal analysis, investigation, data curation, writing—original draft preparation A.S. (Amit Sharma), A.S. (Ashutosh Sharma), P.N., V.G. and H.M.; resources, A.T. and A.B.; writing—review and editing; A.S. (Ashutosh Sharma), A.T., A.B., M.M. and H.M.; Project Supervision A.S. (Ashutosh Sharma), A.T. and A.B.; Funding A.S. (Ashutosh Sharma), M.M. and H.M. All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research, Taif University.

Data Availability Statement: All data in this manuscript has been represented in this manuscript only.

Acknowledgments: The researcher would like to acknowledge the Deanship of Scientific Research, Taif University, for funding this project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xie, Z.; Lv, W.; Huang, S.; Lu, Z.; Du, B.; Huang, R. Sequential graph neural network for urban road traffic speed prediction. *IEEE Access* **2019**, *8*, 63349–63358. [[CrossRef](#)]
2. Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big data analytics in intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 383–398. [[CrossRef](#)]
3. Zulqarnain, M.; Ghazali, R.; Hassim, Y.M.M.; Rehan, M. A comparative review on deep learning models for text classification. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *19*, 325–335. [[CrossRef](#)]
4. Malhi, A.; Yan, R.; Gao, R.X. Prognosis of defect propagation based on recurrent neural networks. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 703–711. [[CrossRef](#)]
5. Khan, A.; Sohail, A.; Zahoor, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [[CrossRef](#)]
6. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 June 2018; pp. 95–104.
7. Chen, Z.; Wang, C.; Li, J.; Xie, N.; Han, Y.; Du, J. Reconstruction bias U-Net for road extraction from optical remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2284–2294. [[CrossRef](#)]
8. Lu, Z.; Lv, W.; Xie, Z.; Du, B.; Huang, R. Leveraging graph neural network with lstm for traffic speed prediction. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI), Leicester, UK, 19–23 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 74–81.
9. Chen, C.; Li, K.; Teo, S.G.; Zou, X.; Wang, K.; Wang, J.; Zeng, Z. Gated residual recurrent graph neural networks for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 2–3 July 2019; Volume 33, pp. 485–492.
10. Fang, Y.; Jiang, J.; He, Y. Traffic Speed Prediction Based on LSTM-Graph Attention Network (L-GAT). In Proceedings of the 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Changsha, China, 26–28 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 788–793.
11. Babu, C.N.; Reddy, B.E. A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data. *Appl. Soft Comput.* **2014**, *23*, 27–38. [[CrossRef](#)]
12. Lv, F.; Wang, J.; Cui, B.; Yu, J.; Sun, J.; Zhang, J. An improved extreme gradient boosting approach to vehicle speed prediction for construction simulation of earthwork. *Autom. Constr.* **2020**, *119*, 103351. [[CrossRef](#)]
13. Priambodo, B.; Jumaryadi, Y. Time series traffic speed prediction using k-nearest neighbour based on similar traffic data. In Proceedings of the MATEC Web of Conferences, Solo Baru, Indonesia, 11–12 July 2018; EDP Sciences: Les Ulis, France; Volume 218, p. 03021.
14. Gopi, G.; Dauwels, J.; Asif, M.T.; Ashwin, S.; Mitrovic, N.; Rasheed, U.; Jaillet, P. Bayesian support vector regression for traffic speed prediction with error bars. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 136–141.
15. Wu, P.; Huang, Z.; Pian, Y.; Xu, L.; Li, J.; Chen, K. A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting. *J. Adv. Transp.* **2020**, *2020*, 8863724. [[CrossRef](#)]
16. Zhang, W.; Yu, Y.; Qi, Y.; Shu, F.; Wang, Y. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transp. A Transp. Sci.* **2019**, *15*, 1688–1711. [[CrossRef](#)]
17. Jiang, W.; Zhang, L. Geospatial data to images: A deep-learning framework for traffic forecasting. *Tsinghua Sci. Technol.* **2018**, *24*, 52–64. [[CrossRef](#)]
18. Guo, J.; Liu, Y.; Yang, Q.; Wang, Y.; Fang, S. GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model. *Transp. A Transp. Sci.* **2021**, *17*, 190–211. [[CrossRef](#)]
19. Wang, H.; Hao, L.; Sharma, A.; Kukkar, A. Automatic control of computer application data processing system based on artificial intelligence. *J. Intell. Syst.* **2022**, *31*, 177–192. [[CrossRef](#)]
20. Sun, L.; Gupta, R.K.; Sharma, A. Review and potential for artificial intelligence in healthcare. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 54–62. [[CrossRef](#)]
21. Cai, Y.; Sharma, A. Swarm intelligence optimization: An exploration and application of machine learning technology. *J. Intell. Syst.* **2021**, *30*, 460–469. [[CrossRef](#)]
22. Shang, C.; Liu, Q.; Chen, K.S.; Sun, J.; Lu, J.; Yi, J.; Bi, J. Edge attention-based multi-relational graph convolutional networks. *arXiv* **2018**, arXiv:1802.04944.

23. Zhu, H.; Koniusz, P. Simple spectral graph convolution. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 2 October 2020.
24. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 2, pp. 729–734.
25. Lu, Z.; Lv, W.; Xie, Z.; Du, B.; Xiong, G.; Sun, L.; Wang, H. Graph Sequence Neural Network with an Attention Mechanism for Traffic Speed Prediction. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–24. [\[CrossRef\]](#)
26. Reza, S.; Ferreira, M.C.; Machado JJ, M.; Tavares, J.M.R. A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. *Expert Syst. Appl.* **2022**, *202*, 117275. [\[CrossRef\]](#)
27. Yuan, H.; Li, G. A survey of traffic prediction: From spatio-temporal data to intelligent transportation. *Data Sci. Eng.* **2021**, *6*, 63–85. [\[CrossRef\]](#)
28. Dong, S.; Wang, P.; Abbas, K. A survey on deep learning and its applications. *Comput. Sci. Rev.* **2021**, *40*, 100379. [\[CrossRef\]](#)
29. Yadati, N. Neural message passing for multi-relational ordered and recursive hypergraphs. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 3275–3289.
30. Zhan, X.; Li, R.; Ukkusuri, S.V. Link-based traffic state estimation and prediction for arterial networks using license-plate recognition data. *Transp. Res. Part C Emerg. Technol.* **2020**, *117*, 102660. [\[CrossRef\]](#)
31. Li, J.; Sun, P.; Hu, Y. Traffic modeling and optimization in datacenters with graph neural network. *Comput. Netw.* **2020**, *181*, 107528. [\[CrossRef\]](#)
32. Niu, K.; Cheng, C.; Chang, J.; Zhang, H.; Zhou, T. Real-time taxi-passenger prediction with L-CNN. *IEEE Trans. Veh. Technol.* **2018**, *68*, 4122–4129. [\[CrossRef\]](#)
33. Hu, Z.; Chen, K.; Xie, X. A Graph Neural Network (GNN) Algorithm for Constructing the Evolution Process of Rural Settlement Morphology. *Secur. Commun. Netw.* **2022**, *2022*, 7517347. [\[CrossRef\]](#)
34. Wang, P.; Shi, T.; He, R.; Yuan, W. Prediction of Large Scale Spatio-temporal Traffic Flow Data with New Graph Convolution Model. *Intell. Electron. Circuits Terahertz ITS Beyond* **2022**, 113–136. [\[CrossRef\]](#)
35. Rajagopal, B.G.; Kumar, M.; Samui, P.; Kaloop, M.R.; Shahdah, U.E. A Hybrid DNN Model for Travel Time Estimation from Spatio-Temporal Features. *Sustainability* **2022**, *14*, 14049. [\[CrossRef\]](#)
36. Yu, J.J.Q.; Gu, J. Real-time traffic speed estimation with graph convolutional generative autoencoder. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3940–3951. [\[CrossRef\]](#)
37. Zhang, L.; Ma, J. A Spatiotemporal Graph Wavelet Neural Network for Traffic Flow Prediction. *J. Inf. Intell.* **2023**, *in press*. [\[CrossRef\]](#)
38. Umare, P.R.; Jayswal, S.G.; Tambakhe, S.R.; Upadhye, P.D.; Gulhane, N.D. Smart Solution for Traffic Control. In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 23–25 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 721–724.
39. Alharbi, A.; Yamin, M.; Halikias, G. Smart Technologies for Comprehensive Traffic Control and Management. In Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 357–362.
40. Zhang, S.; Niu, T.; Wu, Y.; Zhang, K.M.; Wallington, T.J.; Xie, Q.; Xu, H. Fine-grained vehicle emission management using intelligent transportation system data. *Environ. Pollut.* **2018**, *241*, 1027–1037. [\[CrossRef\]](#)
41. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
42. Liu, Y.; Lyu, C.; Zhang, Y.; Liu, Z.; Yu, W.; Qu, X. DeepTSP: Deep traffic state prediction model based on large-scale empirical data. *Commun. Transp. Res.* **2021**, *1*, 100012. [\[CrossRef\]](#)
43. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17804–17815.
44. Wang, X.; Zhu, L.; Zhao, J.; Xu, Z.; Li, Z.; Feng, J.; Deng, C. Adaptive Spatial-Temporal Inception Graph Convolutional Networks for Multi-step Spatial-Temporal Network Data Forecasting. In Proceedings of the ICLR 2021 Conference, Vienna, Austria, 4 May 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.