Tikrit Journal of Pure Science Vol. 30 (2) 2025
DOI: ***https://doi.org/10.25130/tjps.v30i2.1732***

IRAQI Academic Scientific Journals

Tikrit Journal of Pure Science
ISSN: 1813 – 1662 (Print) --- E-ISSN: 2415 – 1726 (Online)
Journal Homepage: https://tjpsj.org/

Tikrit Journal of Pure Science

Using Hybrid Neural Networks to Improve Traffic Prediction and Congestion Management

Ali Abd Samir
Oil Products Distribution Company, Salahuddin Branch, Tikrit, Iraq

Received: 19 Aug. 2024 **Received in revised form:** 8 Oct. 2024 **Accepted:** 13 Oct. 2024
Final Proofreading: 27 Nov. 2024 **Available online:** 25 Apr. 2025

ABSTRACT

Urban regions are commonly plagued by traffic congestion, which results in substantial economic losses and a diminished quality of life. Accurate prediction of traffic flow and effective management of congestion are important in reducing the impacts of traffic. This paper presents a new approach using hybrid neural network models to enhance the accuracy of traffic predictions and improve strategies for congestion management. The proposed Materials and methods integrates Diffusion Convolutional Recurrent Neural Network (DCRNN) with graph-based models, allowing information to be shared among related sensors over large distances. The METR-Los Angeles (METR-LA) dataset consists of traffic data collected from 207 loop detectors located on highways in Los Angeles. Validation is done through various methods that prove the practicality and efficiency of the developed deep learning methodologies for real-time congestion monitoring and management systems.

Keywords: Traffic Prediction, Deep Learning, Graph-Based Models, DCRNN, METR-LA, Congestion Management

Name: **Ali abd Samir** **E-mail:** **lybd3074@gmail.com**

استخدام الشبكات العصبية الهجينة لتحسين التنبؤ المروري وإدارة الازدحام

علي عبد سمير
شركة توزيع المنتجات النفطية فرع صلاح الدين، تكريت، العراق

الملخص

تعاني المناطق الحضرية عادة من الازدحام المروري، مما يؤدي إلى خسائر اقتصادية كبيرة وانخفاض في جودة الحياة. يعد التنبؤ الدقيق بأنماط حركة المرور والتحكم الفعال في الازدحام أمرًا بالغ الأهمية للحد من الآثار السلبية الناجمة عن حركة المرور . تقدم هذه الدراسة طريقة

جديدة تستخدم الشبكات العصبية الهجينة لتحسين دقة التنبؤ بحركة المرور وتعزيز أساليب إدارة الازدحام. يدمج النهج المقترح بشكل تآزري مزايا الشبكات العصبية المتكررة التلافيفية الديناميكية (DCRNN) مع النماذج التقليدية القائمة على الرسم البياني لالتقاط التبعيات المكانية والزمانية الموجودة في بيانات حركة المرور بشكل فعال. تم استخدام مجموعة بيانات METR-LA ، التي تشتمل على قياسات حركة المرور من 207 كاشفًا لحلقات الطرق السريعة في لوس أنجلوس، للتدريب والتحقق من صحة النموذج. تشير النتائج إلى فعالية الشبكات العميقة في إنشاء أنظمة مراقبة وإدارة الازدحام المروري في الوقت الحقيقي.

INTRODUCTION

The enormous increase in the multitude of vehicles has placed a major problem within the urban. Intelligent Transportation Systems (ITS) is in smart cities a wise traffic control system that offers good answers to the problems of city street traffic. This study focuses on the analysis of traffic predictions that are considered large spatial-temporal forecasts. Traffic [(1)] flow refers to different types of motion on the road also involving pedestrians, moving vehicles, and road infrastructure. Traffic flow prediction uses past data related to traffic flow collected by sensors to predict future circumstances [(2)]. This assists people in being able to avoid congestion and choose their routes based on information that would be convenient and safe.

The enormous growth in the number of vehicles has posed a main trouble within the urban. The functioning of traffic forecast as initiation is explained by spatial-temporal predictions considered an autonomous vehicle related model to create vehicle-to-vehicle and infrastructure-to-vehicle trajectories. Intelligent Transportation Systems (ITS) are typically advanced traffic management systems, and this study focuses on its role in smart cities for good solutions to city street traffic issues. ITS will provide custom designs that relate to various types of motion on the street, pedestrian information services (in motion), automotive environmental observations, and road infrastructure.

The tremendous boom in the car population has posed a main issue throughout the urban. Intelligent Transportation Systems (ITS) are synthetic site visitors control systems in smart cities that provide pleasant approaches to city road site visitors issues. This study makes a specialty of traffic forecasting, which is considered huge spatial-temporal prediction. Traffic goes [(1)] refers to the diverse types of movement at the avenue along with pedestrians, automobiles on the cross and street infrastructure space. Traffic Flow Prediction is primarily based on sensor gathered beyond records on traffic glide to make predictions for destiny occasions [(2)]; this will help individuals in preventing congestion and deciding on their routes — via facts which might be each handy and secure.

BACKGROUND

To address this, some approaches learnable node embeddings and base model parameters for the adjacency matrix while others generate the adjacency matrix with a possibility based on. Such an approach allows the network to adapt dynamically depending on the specific details at hand. Further improving over Diffusion Graph Convolutional Recurrent Network (DGCRN), does this by generating a self-adaptive dynamic adjacency matrix at every time step.

The purpose of the spatial topology embedding is to condense the structural facts of the traffic network into properly described graph data preparations, which could be later used for extracting spatial relationship. The original proposal for Convolutional Neural Networks (CNN) based methods [(3)] included dividing maps into grids of equal size and treating them like images, then using convolution to capture relationships between neighboring grids. GNNs provide a more flexible representation of the traffic network, allowing more sophisticated processing of non-Euclidean correlations. The creation of a good adjacency matrix is pivotal to GNNs. A common way is by defining some predefined measure on pairs that indicate proximity between nodes such as geographic distance or connectivity (4). However, this predetermined adjacency matrix is static and has limited ability to capture high variability in spatial relationships among traffic data. To address this problem, some approaches learnable nodes embeddings and use them to produce the adjacency matrix while others treat the adjacency matrix as parameters and implement it by implementing it as part of the model. We consider information at each time step, which makes learning a dynamic graph more reachable—allowing the network to adapt in terms of different traffic conditions. In this way, the Dynamic Graph Convolutional Recurrent Neural Network (DGCRNN) (5) goes one further by learning its own self-adapting dynamic adjacency matrix for each individual point in time.

Although this method provides great flexibility in representing data, the resulting dynamic graphs are not easily understandable in real-world situations. This restricts their capacity to be used to other traffic analysis scenarios.

Recurrent neural networks are commonly used to record temporal dependencies between portions of the sequence. GRU (Gated Recurrent Unit) and LSTM (Long -short-term memory) (6) are further proposed to increase the ability to simulate long-term dependencies by introducing a gate mechanism to control the ratio of preserving long-term information. In traffic prediction, to add spatial information, one intuitive way is to use the outputs of spatial modules as input to Recurrent Neural Network (RNN) Additionally, certain works enhance the computation of gates in GRU and LSTM by including graph convolution (7). In order to decrease the computing expense of RNN, CNN can be used to describe temporal relationships by applying one-dimensional convolution along the time axis (8).

For traffic prediction, spatio-temporal neural networks trained using graph convolution have shown remarkable performance across a variety of tasks, all because to the presence of clearly defined graphs. But it's not easy to predict the relationships between nodes with any degree of certainty due to the complexity of the topology of a real road network. To grasp intricate interdependencies, the authors of (9) suggest an adaptable graph learning method (AdapGL) that is based on convolutional networks. To start, a new graph learning module can be built to adaptively capture additional possible connections between nodes during training. Secondly, the parameters of the prediction network and graph learning modules are optimized using surrogate training, drawing inspiration from the Expectation Maximization (EM) technique. A two-way recurrent neural network turned into created in (10) employing GRUs to extract and categorize traffic as both congested or non-congested. The utilization of real-time data from sensors and related equipment facilitated the more efficient management of traffic. Essential metrics to consider include predicting traffic variables such as velocity, meteorological conditions, current situation, and likelihood of accidents. The performance of congestion prediction has been enhanced by extracting additional information, including traffic, road, and weather conditions. A novel Bayesian framework called Variable Graph Recurrent Attention Neural Networks (VGRAN) is introduced in (11) to enhance the accuracy of traffic prediction. This model uses dynamic graph convolutions to report time-various street sensor signals and analyze latent variables associated with sensor representation and traffic sequences. The suggested probabilistic technique is a versatile generative model that takes into account the stochastic nature of sensor information and the temporal correlations of motion. Furthermore, it allows for effective estimation of differences and precise representation of the fundamental patterns in traffic data, which often exhibit irregularities, geographical correlations, and many temporal aspects.

RELATED WORK

This section compares strategies used to predicate traffic in previous research, that is one of the fundamental tasks inside the field of smart transportation systems Table 1. Statistical methods, artificial intelligence and data mining techniques were used to evaluate road traffic data and predict future traffic indicators. In many studies, machine learning models such as the regression model in [12] have been used to predict traffic data for the next year based on traffic data for previous years. In [13], the prediction accuracy of four ML models was examined using investigation data collected from the road network in Thessaloniki, Greece and the focus was on prediction accuracy and real-time speed. In [14], several algorithms were evaluated to predict traffic flow at an intersection, thus laying the foundation for adaptive traffic control, either by remote control of traffic signals or by implementing an algorithm that adjusts the timing according to the expected flow.

Deep learning and deep neural networks have been used in many studies due to their ability to effectively extract features and handle large volumes of data. In [15], the main objective was to predict trip duration using neural networks such as color clustering algorithm (K-Means algorithm) along with several parameters to calculate and estimate travel duration while using a dataset obtained from Waze Live Map Application Programming Interfaces (APIs). In [16], a Materials and methods for direct traffic status prediction based on spatiotemporal graph using CNN was established. The spatiotemporal graph is fed directly into the traffic prediction model, which uses a CNN. The model was trained using simulated data and a real dataset. However, this study did not investigate the effects of lane changes on the dynamic behavior of traffic flow and prediction accuracy. A traffic situational awareness ensemble technique with a graph implementation on a network of traffic detectors extracted spatial patterns in traffic flow in [17]. After recovering the features, a weight matrix was created to group the underlying models' predictions by performance under given conditions. The efficient real-time traffic flow big data prediction network has important application importance and the main challenge has been how to build an adaptive model based on historical data. Long short-term memory (LSTM) is a special recurrent neural network (RNN) that can learn temporal relationships from sequences of time series due to the memory cells built into it. In [18], LSTM was applied to real-world traffic big data from a benchmark system. In [19], a path-based framework was proposed which can produce better city-scale traffic speed prediction in which the road network is divided into critical paths and each critical path is modeled by Bidirectional Long Short-Term Memory Neural Network (Bi-LSTM NN). In the traffic prediction phase, the spatial and temporal features captured from these processes are fed into a fully connected layer. Finally, the results of each path are aggregated to predict network-level traffic speed. Convolutional Pose Machine (CPM-Conv) LSTM, a spatio-temporal model for short-term prediction of the congestion level on each road segment, was proposed in [20]. The model is built on a spatial matrix that includes both the congestion propagation pattern and the spatial correlation between road segments. In [21], a deep, embedded learning approach (DELA) is proposed that can help, explicitly learn from fine-grained traffic information, road structure and weather conditions. In particular, DELA consists of an embedding component, a CNN component and an LSTM component where the embedding component can capture categorical feature information and identify associated features while the CNN component can recognize 2D traffic flow data while the LSTM component has the benefits of maintaining long-term memory for historical data.

In order to leverage both the spatial and temporal characteristics of traffic data, the researchers initiated the construction of a hybrid model. This involved the integration of two or more distinct models into a single entity. The K-Nearest Neighbors (KNN) LSTM model in [22] uses both spatial data through the selection of the most relevant neighbor and temporal variability to accurately predict the flow. In [23], an Autoencoder-LSTM fusion model was used to capture the internal relationship of traffic flow using an Autoencoder. The LSTM network was then employed to forecast the complex linear traffic flow. The researcher introduced a new model named Letter of Credit (LC) RNN in [24] to forecast road traffic speed. This model comprises a look-up convolutional layer and recurrent layers. The look-up operation retrieves all the neighboring road segments, the convolutional operation captures the spatial relationships, and the recurrent layers acquire the long-term temporal patterns. In addition, the researcher introduced a deep learning model called SCRN in [25], which combines CNN and LSTM. Initially, CNN analyses the spatial characteristics of the traffic network for each time period. Subsequently, the LSTM network acquires knowledge of the temporal relationship in the time-series data to forecast the speed of 278 road links.

The hybrid neural network outperforms both simple neural networks and traditional approaches by effectively extracting spatial and temporal information from the traffic data. Despite the promising outcomes of the hybrid model in traffic prediction, there is a scarcity of research on traffic congestion prediction using deep neural networks. This is mostly because there is a lack of reliable city-wide congestion data. The Autoencoder model was trained by the researcher in [26] using artificially compressed samples of traffic photos obtained from an open-source website. The purpose of the training was to anticipate traffic congestion. The anticipated photographs lack visual intuitiveness due to significant loss of road information during image reduction. In [27], the researchers used bus driving time data during peak periods in order to train the LSTM network to forecast the duration of traffic congestion on six specific road segments. The researchers in [28] employed machine learning algorithms (logistic regression, random forest, and neural networks) on vehicle trajectory data obtained via connected car technologies to detect and forecast the occurrence of traffic congestion. The study contains prediction horizons of (10 and 20 sec), specifically designed to alert drivers of imminent traffic conditions. In [29], the city-level transport picture data from TOPIS was used to forecast city-level traffic congestion in the short and medium term using a hybrid architecture that incorporates CNN, LSTM, and transposed CNN.

The rise of hybrid deep neural network approaches paved way for the study of alternative architectures that could also be able to learn graph-like structure data, which ultimately led to formulation of Graph Neural Networks (GNNs). GNNs are an entirely new kind of deep learning algorithms appropriate for tasks with graph data structures. With respect to this task, it is very important to not only capture the spatial graph-like structure of road networks but also the temporal dimension and its corresponding information. The idea was first introduced by Yu et al. in 2018 when they proposed their Spatio-Temporal Graph Convolutional Neural Network. It integrates graph convolutions and gated temporal convolutions in the use spatio-temporal convolutional blocks [8].

Compared to other techniques to traffic prediction, this model performed far better. As a result, it paved the path for other researchers to investigate these networks and served as a baseline model for comparisons in future works. In [30], a different GNN method was used a residual recurrent structure to effectively capture spatial dependencies and temporal dynamics inside graphs. Specifically, it focused on identifying periodic temporal correlations. GNN model was introduced in [31], which includes external factors. The first convolution layer of the graph captures the spatial correlations, and this is followed by subsequent convolution layers that learn the temporal dynamics. The merging module considers both social factors and road infrastructure. The occurrence of accidents can affect traffic predictions, so a model was developed to capture the impact of traffic accidents on traffic flow and speed in [32], and show the extent to which this affects predictions. Another architecture, known as the encoder-decoder, is proposed and used in reference [3]. This architecture comprises an encoder and a decoder consisting of numerous spatiotemporal attention blocks that capture the influence of spatio-temporal elements on traffic conditions. The encoder analyses the input traffic characteristics, while the decoder generates the output sequence. An intermediary attention layer, known as a transform attention layer, is positioned between the encoder and decoder. Its purpose is to turn the encoded traffic features into the input required by the decoder. The transformer attention technique is designed to capture the direct connections between previous and future time steps, hence reducing the problem of error propagation between prediction time steps. Another approach, described in [33], uses Recurring Gates Units (GRU) to overcome the constraints of GCN in capturing global spatial correlations. This method leverages GRU and focusses on simultaneously analyzing local and global temporal correlations. To address the neglect of node properties, a dynamic spatio-temporal graph convolutional network (DSTGCN) was proposed in [34], which includes a dynamic graph generation module that adaptively integrates geographic proximity and spatial heterogeneity information, and a graph convolutional cycle module that captures local temporal dependencies.

Table 1: Comparison of previous studies in traffic forecasting.

Ref.	Dataset	Method	Findings
(10)	Two Kaggle datasets. Some 2016 data includes date, time, number of cars, and number of intersections and 2017 traffic data.	Regression Mode	More traffic management elements should be considered.
(11)	Data from the road network in Thessaloniki, Greece.	Random Forest (RF), Support Vector Regression (SVR), Multilayer Perceptron (MLP), and Multiple Linear Regression (MLR)	The SVR model works best in stable settings with little changes, whereas the MLP model adapts better to greater changes and has the fewest mistakes.
(12)	A Road Traffic Prediction Dataset from the Huawei Munich Research Center.	Linear Regression, MLP Regressor, Gradient Boosting Regressor, RF Regressor, and Stochastic Gradient Descendent Regressor	Multilayer Perceptron Neural Network obtained better results but took less time to train.
(13)	Dataset obtained using Waze Live Map APIs.	K-Means algorithm	Factors such as weather conditions were not considered.
(14)	Simulated data and a real-world dataset	CNN	Predicting traffic conditions based on time and space diagram.
(15)	Caltrans PeMS dataset.	Support Vector Regression (SVR), Long Short-term Memory (LSTM)	There is a need to improve the network architecture and parameter choices.

(16)	Real-world traffic big data of PeMS.	Long Short-term Memory (LSTM)	The training time needs to be regulated and the number of optimized parameters needs to be expanded.
(17)	Automated vehicle identification detectors data in the core area of Xuancheng, China.	Bidirectional Long Short-Term Memory Neural Network (Bi-LSTM NN)	The model was reasonable and interpretable.
(18)	Data of Helsinki, Finland collected using HERE Traffic API.	ConvLSTM	Places of interest, weather, and the environment should be considered.
(19)	Traffic flow information for approximately 3 months provided by KDD CUP 2017.	CNN and LSTM	A limited learning ability of the embedded component.
(20)	Real-time traffic flow data provided by the Transportation Research Data Lab (TDRL) at the University of Minnesota Duluth (UMD) Data Center.	KNN-LSTM model	The proposed model can achieve an accuracy improvement of 12.59% on average.
(21)	Caltrans PeMS dataset.	AutoEncoder Long Short-Term Memory (AE-LSTM)	Mean Relative Error MRE was reduced by 0.01.
(22)	Two datasets from Beijing and Shanghai.	LC-RNN	The fusion with other information, including periodicity and context factors, is also considered to further improve accuracy.
(23)	Beijing transportation network with 278 links.	Spatiotemporal Recurrent Convolutional Networks (SRCNs)	The spatial dependencies can be captured by DCNNs, and the temporal dynamics can be learned by LSTMs.
(24)	A dataset was created based on traffic congestion map snapshots from the Washington State Department of Transportation's traffic service provider.	Autoencoder-based neural network mode	Photographs lack visual intuitiveness due to the loss of road information during image reduction.
(25)	A total of 66,228 bus driving records were collected from 50 buses over 66 working days in Guangzhou, China.	Method based on bus driving time (TCP-DT) and LSTM	The method can provide a driving path with the least congestion time.
(26)	Vehicle trajectory data obtained via connected car technologies.	Logistic Regression, Random Forest, and Neural Network	10- and 20-second predictive horizons were obtained.
(27)	Caltrans PeMS dataset.	CNN, LSTM, and Transpose-CNN	More data from multiple sources is needed for more accurate forecasts.
(28)	Two datasets: BJER4 and PeMSD7.	Spatio-Temporal Graph Convolutional Networks (STGCN)	The model achieves fast training and fewer parameters.
(29)	Two datasets: METR-LA and PEMS-BAY.	Residual Recurrent Graph Neural Networks (Res-RGNN)	Spatial and temporal features should be investigated for better interpretation.
(30)	Two Caltrans PeMS datasets (PEMSD7 and PEMSD4).	Temporal Graph Convolutional Networks (GTCN)	MAE: 0.64

[31]	Two real-world urban traffic datasets of San Francisco and New York city.	Deep Incident-Aware Graph Convolutional Network (DIGC-Net)	Effectiveness in extracting accident features.
[32]	Two datasets: Xiamen and PeMS.	Graph Multi-Attention Network (GMAN)	4% improvement in MAE.
[33]	METR-LA and PeMS-Bay	Graph Convolutional Recurrent Attention Network (GCRAN)	Extract local and global spatial correlations simultaneously.
[34]	PEMS-Bay, NE-BJ, PEMSD4, PEMSD8	Dynamic Spatial-Temporal Graph Convolutional Network (DSTGCN)	Prediction of flow and speed.

Anticipating and forecasting traffic flows is crucial due to the potential risks of traffic congestion, particularly in densely populated areas. Consequently, there is a requirement for practical and efficient road traffic prediction methods. Key issues are around the absence of computationally streamlined approaches and algorithms. Furthermore, there are constraints when it comes to obtaining training data of superior quality. The lack of use of dynamically acquired spatio-temporal correlations in deep learning is a significant challenge, as it fails to account for the intricate relationship between road sections and traffic congestion patterns or crowded areas.

MATERIALS AND METHODS

A hybrid deep neural network is proposed for traffic forecasting by predicting future road speeds based on previous speeds measured by the sensor at the same location using a hybrid model that combines two types of models.

- First, distinguish between basic road characteristics and sensor locations along the path using the unique ability of convolutional networks to extract features and use graphical input provided via the network.
- Second, transfer the output to recursive neural networks, which are known for their ability to work with sequential data to predict future speeds.

Figure 1 shows the outline of the Materials and methods to be followed. The raw traffic time series data is processed and transformed into graph-structured data and then the graph data is fed into a hybrid deep learning model that handles temporal and spatial dependencies. The proposed model is trained and its performance in traffic prediction is evaluated.

Fig. 1: Proposed Materials and methods for traffic forecasting.

The primary programming language used is Python. This was chosen because of the wide usage and popularity plus the availability of numerous libraries for data analysis and machine learning tasks. Python is a well-known programming language for machine learning of research because of its large amount and focused community of developers and users, in addition to its user-friendly nature, easy syntax [35].

The programming language has an extensive collection of libraries and tools specifically created for machine learning. These resources offer a diverse selection of algorithms, models, and tools for processing data. Python is a versatile and scalable programming language that can efficiently handle extensive datasets and intricate models. Moreover, it offers compatibility with many languages and technologies, enabling the combination of their respective capabilities. The

DOI: https://doi.org/10.25130/tjps.v30i2.1732

prediction algorithm was implemented on an NVIDIA T4 GPU with a 585MHz GPU which can be increased to 1590MHz and also contains 2560 NVIDIA cores and RAM capacity is 16 MB.

Dataset

The METR-LA dataset, consisting of 207 loop detectors, gives information on the flow and occupancy observed from the Los Angeles County Road network freeway. The data had been collected at five-minute intervals [36].

The predominant time frame in this dataset spans from March 1 to June 30, 2012. Figure 2 displays the positions of detectors in the road network using red pins. The dataset uses a non-directional graph with edge weights to create the adjacency matrix.

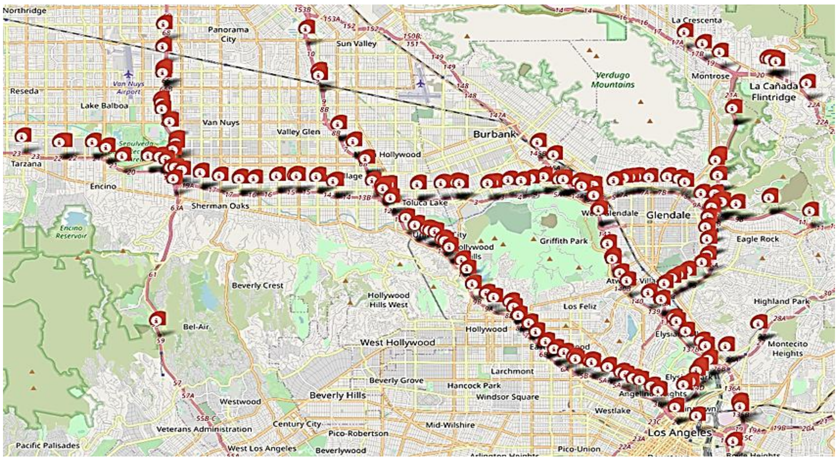

Fig. 2: Locations of loop detectors in METR-LA dataset [36].

The distances between detectors are calculated pairwise, and then an adjacency matrix is constructed using a thresholded Gaussian Kernel based on the method described [37]. The edge weights are determined using the equation 1 provided below:

$$W_{i,j} = e^{-\frac{dist\ (i,j)^2}{2Q^2}} \quad if\ \ dist(i,j) < d_{thresholed}\ ,0\ otherwize \qquad \ldots (1)$$

Where the real physical road distance between nodes i and j in the road network is represented by dist (i, j), and Wi,j is the edge weight between nodes i and j. The initial distance is represented by dthreshold, and the standard deviation of those distances is σ. Figure 3 shows the structure of the data set where each row represents a 5-minute interval, and each column represents a sensor. The value in each cell corresponds to the average harmonic speed for that period, which is measured in miles per hour (mph).

AGG_PERIOD_START	400001	400017	400030	400040	400045	400052	400057	400059	400065	...	413877	413878	414284	414694
2017-01-01 00:00:00	71.4	67.8	70.5	67.4	68.8	66.6	66.8	68.0	66.8	...	70.4	68.8	71.1	68.0
2017-01-01 00:05:00	71.6	67.5	70.6	67.5	68.7	66.6	66.8	67.8	66.5	...	70.1	68.4	70.8	67.4
2017-01-01 00:10:00	71.6	67.6	70.2	67.4	68.7	66.1	66.8	67.8	66.2	...	69.8	68.4	70.5	67.9
2017-01-01 00:15:00	71.1	67.5	70.3	68.0	68.5	66.7	66.6	67.7	65.9	...	70.2	68.4	70.8	67.6
2017-01-01 00:20:00	71.7	67.8	70.2	68.1	68.4	66.9	66.1	67.7	66.1	...	70.0	68.4	71.0	67.9

Fig. 3: Structure of the METR-LA dataset.

After downloading and analyzing the data set, it was found that each sample contains data for 207 nodes with two features for each node (speed and time) across 12-time steps. The goal is to predict the normalized velocity for the next 12-time steps for each node, and edges are identified based on the distances between threshold sensors. Figure 4 displays a single data point that mimics the structure of the data set and prints the values of the first sample. where x are the features (207 nodes, 2 features per node, 12-time steps) and edge labels with the shape indicate connections between nodes) and there are edge features and labels y of the shape [207, 12] (the normalized speed of 207 nodes for the next 12-time steps).

```
Data(x=[207, 2, 12], edge_index=[2, 1722], edge_attr=[1722], y=[207, 12])
```

Fig. 4: First sample.

Data Processing

The time series forecasting problem can be considered a supervised learning problem. We can do this by using the previous time steps as input features and using the next time step as the output for the prediction. Then, the question of spatio-temporal prediction can be formulated as predicting the value of a feature in the future, given the historical values of the feature for that entity as well as the feature values of entities "connected" to the entity. For example, in a speed prediction problem, the historical speeds of sensors are time sequences and the distance between sensors is an indicator of connectivity or proximity to the sensors.

The first stage involves collecting traffic data and importing it into the system using a specialized data loader. In order to ensure stable and effective training, the data undergoes Z-score normalization, which standardizes the data by adjusting it to have a mean of 0 and a uniform deviation of 1. Next, the data undergoes conversion to a histogram format, where the adjacency matrix depicts the correlation between sensors. This graph represents the spatial connections between sensors.

In order to generate the training dataset, sequences of traffic data are retrieved as both features and targets. The feature sequences comprise a predetermined number of input time steps, while the target sequences encompass the succeeding output time steps. This is further refined by a specialized processing function that prioritizes the speed aspect and modifies the data to align with the input specifications of the model. The completed dataset, which includes these characteristics such as features, targets, edges, and edge weights, is organized into a Static Graph Temporal Signal object. The pre-processing pipeline guarantees the efficient conversion of raw traffic data into a structured format that is appropriate for training the model, enabling precise prediction of traffic patterns.

Modeling

Graph neural networks have proven to be an effective solution for predicting traffic tasks due to their ability to capture the complex spatio-temporal dependencies of traffic data. In the past few years, different types of graph neural networks have been developed, one of which is graph convolutional networks (GCN)[38]. GCNs perform similar operations to CNNs where the model learns features by examining neighboring nodes. The main difference as shown in Figure 5 is that CNNs are specifically designed to work on regular structured data, while GNNs are the generalized version of CNNs where the number of node connections varies and the nodes are unordered.

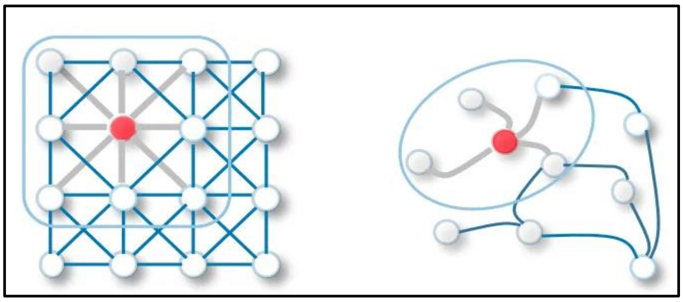

Fig. 5: 2D Convolutional Neural Networks (left) and Graph Convolutional Networks (right).

RNNs excel at processing time-dependent data by retaining the state or memory of prior inputs. RNNs are specifically built to handle sequential data, where the input order has significance. This renders it appropriate for tasks like as time series prediction, speech identification, language modelling, and other similar applications. RNNs incorporate internal loops that enable the retention of information. This allows the network to recall past inputs and utilize that information during the processing of the current input.

The Convolutional Recurrent Neural Network (DCRNN) efficiently integrates the advantages of GCNs and RNNs for processing spatiotemporal data. Figure 6 depicts the network architecture, which is segregated into encryption and decoding components [(39)]. The model begins by utilizing time-series input graph signals, which depict data across a temporal dimension. The encoder comprises convolutional recurrent layers that are propagated. The geographical dependencies in these layers are captured using graph convolutions that consider bidirectional information flow, while the temporal dependencies are captured using recurrent units. Following each propagated recurrent convolutional layer, the Rectified Linear Unit (ReLU) activation function is used to introduce nonlinearity and enhance the learning capability. The encoder analyses the input sequence and advances the hidden states in a forward direction across time, efficiently condensing the temporal information up to the present time step. After the encryption phase concludes, the hidden states are duplicated in order to initialize the decoder. This transition guarantees that the temporal context, which is collected by the encoder, is accessible during the decoding phase. The decoder, like the encoder, is comprised of convolutional recurrent layers that utilize the hidden states of the encoder to forecast future histogram signals. Additionally, the decoder applies ReLU activation after each layer to maintain nonlinearity in the decoding process. The decoder generates forecasts for future time intervals by using the acquired spatiotemporal relationships. The ultimate result comprises projected graph signals for next time intervals, encompassing spatial and temporal patterns in the data.

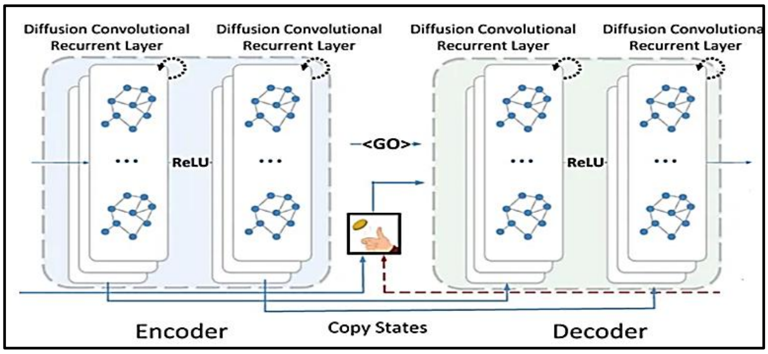

Fig. 6: DCRNN structure [39].

DCRNN effectively integrates GCNs and RNNs to process spatiotemporal data. It uses diffusion convolution to capture spatial dependencies and recurrent layers to simulate temporal dynamics. While it does not incorporate conventional CNNs, it applies convolution procedures to graph data. DCRNN is highly effective in situations like traffic prediction, where spatio-temporal patterns play a vital role. Figure 7 depicts the suggested model design of a graph-based temporal convolutional network, which aims to enhance traffic prediction by utilizing the advantages of graph convolutional networks (GCNs) and RNNs, notably the diffuse convolutional recurrent neural network (DCRNN).

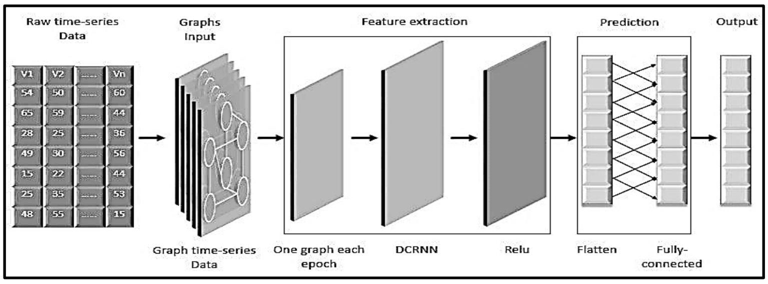

Fig. 7: Proposed model layers.

- Raw Time-Series Data: Provide input data that represents the traffic readings obtained from sensors over a specific period of time. Every row represents a sensor measurement taken at various points in time.
- Graph Time-Series Data: The unprocessed time series data is converted into a sequence of graphs, with nodes representing sensors and edges representing the connections between them. This transformation enables the model to incorporate the spatial relationships between sensors.
- Graph Input: The model is fed with time series data in the form of graphs, with each graph representing the traffic circumstances at a particular time. The model analyses and handles one graph throughout each time period, allowing it to acquire knowledge about temporal patterns spanning many time periods.
- Feature Extraction: The DCRNN layer is specially designed to detect both the temporal and geographical dependencies found in traffic information. The system analyses the incoming network data over a period of time, considering the interactions between nodes (sensors). DCRNN integrates the concepts GCNs with RNNs. The GCN module processes spatial interactions by graph convolutions, while the RNN module handles temporal dependencies by preserving the hidden state over time. This layer generates a transformer sequence by updating the characteristics of each node based on its own historical data and the data from its neighboring nodes.

DOI: https://doi.org/10.25130/tjps.v30i2.1732

- Activation Function: Rectified Linear Unit (ReLU) activation feature introduces nonlinearity into the model, so allowing it to acquire a deeper understanding of intricate patterns. It aids in capturing non-linear correlations in data, which are frequently seen in real-world traffic situations. ReLU function returns the input value as is if it is positive; else, it returns zero. This straightforward procedure helps mitigate the issue of gradient vanishing, hence facilitating the training of deep neural networks.
- Flattening Layer: The flattening layer converts the output, which is in multiple dimensions, into a tensor that has just one dimension. This transformation is essential for establishing a connection between the recurrent layers and the fully linked (linear) layer.
- Fully Connected Layer: It bridges the extracted features, which are in a high-dimensional space, to the final output space. This mapping involves a linear transformation of the input data using the weights and biases that have been learned. This transformation integrates data from all preceding layers to get the ultimate forecast.
- Output: Represents the expected traffic conditions for the next time steps for traffic management and congestion forecasting.

Figure 8 depicts the hierarchical structure of the proposed model, which is a sophisticated neural network design that combines a Diffusion Convolutional Recurrent Neural Network (DCRNN) layer with multiple subsequent processing stages. The initial element, DCRNN-1, comprises of an LSTM layer followed by a linearization layer. The LSTM has an input dimension of 207 and a hidden dimension of 250, leading to a substantial parameter count of 125,000. The weights and biases used in LSTM gates and recurrent connections are represented by it. Furthermore, the LSTM linear layer generates a single value for every time step, contributing 251 parameters to the model, encompassing both weights and biases. After passing through the LSTM and linear layers, the output is processed by the ReLU activation function. Next, the output from a 3D tensor [1, 207, 1] is normalized to a 2D tensor [1, 207], preparing it for the final fully connected layer. The final stage is the linear layer, which performs the transformation from 207 features to 207 output features. This layer provides 42,441 parameters, calculated from the product of input and output sizes plus biases. This linear transformation is crucial for mapping the high-dimensional representations learned by the network to the desired output space.

Layer (type)	Output Shape	Param #
DCRNN-1	[1, 207, 1]	125,250
LSTM	[1, 207, 250]	125,000
Linear	[1, 207, 1]	251
ReLU-2	[1, 207, 1]	N/A
Flatten-3	[1, 207]	N/A
Linear-4	[1, 207]	42,441

Total params: 167,691
Trainable params: 167,691
Non-trainable params: 0

Fig. 8: Hierarchical structure of the proposed model layers.

Training

The dataset was partitioned between training and testing subsets using an 80/20 ratio, guaranteeing that the model was trained on one portion and validated on another. The data set contains 34,249 samples, of which 27,408 samples were allocated for training and 6,852 samples for testing. The loss function employed was the mean squared error Mean Squared Error (MSE), which quantifies the average squared difference between the projected and actual traffic speeds. The Adam optimizer was employed to update the model parameters, reducing the loss function by modifying the weights during backpropagation. The training process is iterated over numerous epochs, wherein the model at each epoch processes the training data, computes the loss, and adjusts the weights. This method is iterated till the model attains the most reliable solution.

All through each training generation, the loss is saved and monitored so as to tune the model's learning progress. These metrics were visually depicted in real-time using Tensor Board, giving an insight into whether the model can memorize rather than learn from a pattern. Visualization across epochs of loss values is also implemented for spotting patterns and possible problems to do with vanishing or bursting gradients, and making sure where the model starts converging. This perception ensures that any irregularities in the learning process are detected immediately and resolved. Figure 9 Training logs show a very great improvement as regards model performance: MSE values decrease progressively from one epoch to another. The hybrid neural network successfully captures both the temporal and spatial connections in traffic data, resulting in precise traffic predictions.

```
  5%|▌          | 1/20 [03:33<1:07:28, 213.10s/it]MSE: 0.9379
 10%|█          | 2/20 [07:07<1:04:11, 213.97s/it]MSE: 0.8257
 15%|█▌         | 3/20 [10:58<1:02:45, 221.50s/it]MSE: 0.6946
 20%|██         | 4/20 [14:35<58:36, 219.80s/it]  MSE: 0.5934
 25%|██▌        | 5/20 [18:19<55:22, 221.48s/it]MSE: 0.5458
 30%|███        | 6/20 [21:55<51:13, 219.53s/it]MSE: 0.5310
 35%|███▌       | 7/20 [25:46<48:21, 223.19s/it]MSE: 0.5128
 40%|████       | 8/20 [29:21<44:09, 220.76s/it]MSE: 0.4908
```

Fig. 9: Training logs.

The plot depicted in Figure 10 facilitates the visual assessment of the model's performance by comparing the projected traffic speeds with the actual traffic speeds. The x-axis depicts the temporal intervals in the dataset, spanning from around (0 to 7000). The y-axis indicates traffic speeds that have been normalized and standardized using Z-score normalization. The range of normalized speeds is from (-2.5 to 1.0). The blue line depicts the traffic speeds forecasted by the model, while the orange line reflects the traffic speeds actually measured by the sensors. The model effectively reflects overall patterns in actual traffic speeds and somewhat captures significant decreases in speeds, which are indicative of traffic congestion. This suggests that the model possesses a certain level of capability in forecasting congestion events, and the strong agreement between the anticipated speeds and actual speeds for the majority of time intervals indicates that the model is working adequately in its present condition.

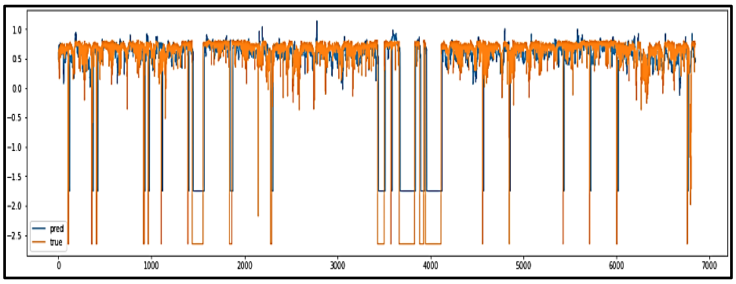

Fig. 10: Comparing the predicted traffic speeds to the true traffic speeds.

Evaluation

Following the training process, the model is assessed on the test dataset to determine its capacity for generalization after training the model in one hour. MAE measures the average magnitude of errors in a set of predictions, without considering their direction. Root Mean Squared Error (RMSE) evaluates the square root of the average squared differences between predicted and actual values, emphasizing larger errors. The model is configured to operate in evaluation mode, guaranteeing that layers such as dropout and batch normalization function correctly during the inference process. The evaluation loop iterates through the test dataset, where for each snapshot, the model makes predictions about node attributes based on the input

DOI: https://doi.org/10.25130/tjps.v30i2.1732

graph's structure. The predictions then compared to the actual values with a view to calculate the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) step by step. The labels and actual predictions are then added to their appropriate lists to assist with additional statistical calculations. Upon completion of the processing of the whole test data set, the average MAE and RMSE are computed by dividing the accumulated mistakes by the number of shots. These average values are then converted into numerical values for simpler reading, as illustrated in Figure 11.

```
MAE: 0.4751
RMSE: 0.6806
```

Fig. 11: Evaluation results.

The proposed model demonstrates notable performance, attaining an Mean Absolute Error (MAE) of (0.4751) and RMSE of (0.6806). This surpasses the performance of the original models presented in Table 2, so confirming the enhanced model's efficacy in delivering more precise traffic forecasts. These models play a crucial role in using deep learning to comprehend and forecast intricate spatiotemporal patterns, rendering them highly effective in domains like transportation management and urban planning. STGCN (Spatio-temporal Graph Convolutional Network) is specifically developed to process spatio-temporal graph data. ASTGCN (Attention-Based Spatio-temporal Graph Convolutional Network) improves upon STGCN by integrating attention processes. This enables the model to preferentially concentrate on significant nodes and edges within the graph structure. GCRNN (Graph Recurrent Neural Network) is a model that integrates graph convolutional layers with recurrent neural networks, much like DCRNN.

Table 1: Comparing the performance of the proposed prediction model with other models.

	Proposed Model	DCRNN	STGCN	ASTGCN	GCRNN
MAE	0.4751	3.60	4.59	6.51	3.70
RMSE	0.6801	7.59	9.40	12.52	8.16

The improved performance was attributed to the integration of linear layers, which improved the model's ability to capture underlying patterns in traffic data. The model is stored for future utilization, enabling its deployment in real-time traffic management systems. When it comes to improving the accuracy of traffic float estimates, hybrid solutions have typically been proven to be helpful. These approaches are able to represent the complexities of traffic patterns and the impact of external variables because they combine the benefits of many models. It should be mentioned that hybrid approaches have the potential to enhance traffic flow predictions, but they could also be more expensive to compute and necessitate more data and resources for maximum performance.

CONCLUSION

Accurate traffic forecasting and congestion control are now critical elements of city planning and transportation systems. These practices enable more informed decision-making, cut travel time, and enhance overall road safety. Conventional approaches frequently struggle to accurately represent the intricate spatial and temporal patterns that are inherent in traffic data. A novel hybrid neural network model was created and assessed to enhance the precision of traffic prediction and facilitate the implementation of efficient congestion management strategies. This was achieved by integrating DCRNN with conventional graph-based models, enabling the model to effectively capture both spatial and temporal dependencies in predictive traffic data. The METR-LA dataset was used for both training and validating the model. The hybrid model exhibited reduced MAE and RMSE compared to conventional techniques in traffic forecasting, demonstrating superior predicted accuracy and dependability. An important contribution of this research is to showcase the efficacy of the hybrid approach in facilitating real-time traffic management systems. Precise and accurate traffic predictions empower transportation authorities to proactively implement steps to alleviate congestion, thereby enhancing traffic movement and diminishing the related economic and environmental burdens. Future efforts will concentrate on enhancing the model to a greater extent, investigating additional datasets, and incorporating real-time data sources to augment the predictive powers and practical utility of the hybrid neural network model.

Conflict of interests: The author declared no conflicting interests.

Sources of funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author contribution: The author made the primary contribution to the study.

REFERENCES

1. Yuan H, Li G. A survey of traffic prediction: from spatio-temporal data to intelligent transportation. Data Science and Engineering. 2021 Mar; 6(1):63-85. https://doi.org/10.1007/s41019-020-00151-z

2. Fang Z, Pan L, Chen L, Du Y, Gao Y. MDTP: A multi-source deep traffic prediction framework over spatio-temporal trajectory data. Proceedings of the VLDB Endowment. 2021 Apr 1;14(8):1289-97. https://doi.org/10.14778/3457390.3457394

3. Zheng C, Fan X, Wang C, Qi J. Gman: A graph multi-attention network for traffic prediction. InProceedings of the AAAI conference on artificial intelligence 2020 Apr 3 (Vol. 34, No. 01, pp. 1234-1241). https://doi.org/10.1609/aaai.v34i01.5477

4. Zhang Q, Chang J, Meng G, Xiang S, Pan C. Spatio-temporal graph structure learning for traffic forecasting. InProceedings of the AAAI conference on artificial intelligence 2020 Apr 3 (Vol. 34, No. 01, pp. 1177-1185). https://doi.org/10.1609/aaai.v34i01.5470

5. Lin Z, Feng J, Lu Z, Li Y, Jin D. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. InProceedings of the AAAI conference on artificial intelligence 2019 Jul 17 (Vol. 33, No. 01, pp. 1020-1027). https://doi.org/10.1609/aaai.v33i01.33011020

6. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H. T-GCN: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems. 2019 Aug 22;21(9):3848-58. https://doi.org/10.1109/TITS.2019.2935152

7. Bai L, Yao L, Li C, Wang X, Wang C. Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems. 2020;33:17804-15. https://doi.org/10.48550/arXiv.2007.02842

8. Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875. 2017 Sep 14. https://doi.org/10.48550/arXiv.1709.04875

9. Zhang W, Zhu F, Lv Y, Tan C, Liu W, Zhang X, Wang FY. AdapGL: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks. Transportation Research Part C: Emerging Technologies. 2022 Jun 1;139:103659. http://dx.doi.org/10.1016/j.trc.2022.103659

10. Abdullah SM, Periyasamy M, Kamaludeen NA, Towfek SK, Marappan R, Kidambi Raju S, Alharbi AH, Khafaga DS. Optimizing traffic flow in smart cities: Soft GRU-based recurrent neural networks for enhanced congestion prediction using deep learning. Sustainability. 2023 Mar 29;15(7):5949. https://www.mdpi.com/2071-1050/15/7/5949#

11. Zhou F, Yang Q, Zhong T, Chen D, Zhang N. Variational graph neural networks for road traffic prediction in intelligent transportation systems.

IEEE Transactions on Industrial Informatics. 2020 Jul 14;17(4):2802-12. https://doi.org/10.1109/TII.2020.3009280

12. Deekshetha HR, Shreyas Madhav AV, Tyagi AK. Traffic prediction using machine learning. In Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2021 2022 Mar 22 (pp. 969-983). Singapore: Springer Singapore. http://dx.doi.org/10.1007/978-981-16-9605-3_68

13. Bratsas C, Koupidis K, Salanova JM, Giannakopoulos K, Kaloudis A, Aifadopoulou G. A comparison of machine learning methods for the prediction of traffic speed in urban places. Sustainability. 2019 Dec 23;12(1):142. https://www.mdpi.com/2071-1050/12/1/142#

14. Navarro-Espinoza A, López-Bonilla OR, García-Guerrero EE, Tlelo-Cuautle E, López-Mancilla D, Hernández-Mejía C, Inzunza-González E. Traffic flow prediction for smart traffic lights using machine learning algorithms. Technologies. 2022 Jan 10;10(1):5. https://doi.org/10.3390/technologies10010005

15. Pangesta J, Dharmadinata OJ, Bagaskoro AS, Hendrikson N, Budiharto WI. Travel duration prediction based on traffic speed and driving pattern using deep learning. ICIC Express Lett Part B Appl. 2021 Jan;12(1):83-90. https://DOI:10.24507/icicelb.12.01.83

16. Bao X, Jiang D, Yang X, Wang H. An improved deep belief network for traffic prediction considering weather factors. Alexandria Engineering Journal. 2021 Feb 1;60(1):413-20. https://doi.org/10.1016/j.aej.2020.09.003

17. Chen Y, Lv Y, Ye P, Zhu F. Traffic-condition-awareness ensemble learning for traffic flow prediction. IFAC-Papers OnLine. 2020 Jan 1;53(5):582-7. https://doi.org/10.1016/j.ifacol.2021.04.146

18. Kong F, Li J, Jiang B, Zhang T, Song H. Big data-driven machine learning-enabled traffic flow prediction. Transactions on Emerging Telecommunications Technologies. 2019 Sep;30(9):e3482. https://doi.org/10.1002/ett.3482

19. Wang J, Chen R, He Z. Traffic speed prediction for urban transportation network: A path based deep learning approach. Transportation Research Part C: Emerging Technologies. 2019 Mar 1;100:372-85. https://doi.org/10.1016/j.trc.2019.02.002

20. Di X, Xiao Y, Zhu C, Deng Y, Zhao Q, Rao W. Traffic congestion prediction by spatiotemporal propagation patterns. In2019 20th IEEE international conference on mobile data management (MDM) 2019 Jun 10 (pp. 298-303). IEEE. https://doi.org/10.1109/MDM.2019.00-45

21. Zheng Z, Yang Y, Liu J, Dai HN, Zhang Y. Deep and embedded learning approach for traffic flow prediction in urban informatics. IEEE Transactions on Intelligent Transportation Systems. 2019 Apr 22;20(10):3927-39. http://dx.doi.org/10.1109/TITS.2019.2909904

22. Luo X, Li D, Yang Y, Zhang S. Spatiotemporal traffic flow prediction with KNN and LSTM. Journal of Advanced Transportation. 2019;2019(1):4145353. https://doi.org/10.1155/2019/4145353

23. Wei W, Wu H, Ma H. An autoencoder and LSTM-based traffic flow prediction method. Sensors. 2019 Jul 4;19(13):2946. https://www.mdpi.com/1424-8220/19/13/2946#

24. Lv Z, Xu J, Zheng K, Yin H, Zhao P, Zhou X. Lc-rnn: A deep learning model for traffic speed prediction. InIJCAI 2018 Jul 13 (Vol. 2018, p. 27). http://dx.doi.org/10.24963/ijcai.2018/482

25. Yu H, Wu Z, Wang S, Wang Y, Ma X. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors. 2017 Jun 26;17(7):1501. https://doi.org/10.3390/s17071501

26. Zhang S, Yao Y, Hu J, Zhao Y, Li S, Hu J. Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks. Sensors. 2019 May 14;19(10):2229. https://doi.org/10.3390/s19102229

27. Huang Z, Xia J, Li F, Li Z, Li Q. A peak traffic congestion prediction method based on bus driving time. Entropy. 2019 Jul 19;21(7):709. https://doi.org/10.3390/e21070709
28. Elfar A, Talebpour A, Mahmassani HS. Machine learning approach to short-term traffic congestion prediction in a connected environment. Transportation Research Record. 2018 Dec; 2672(45):185-95. https://doi.org/10.1177/0361198118795010
29. Ranjan N, Bhandari S, Zhao HP, Kim H, Khan P. City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN. Ieee Access. 2020 Apr 30;8:81606-20. https://doi.org/10.1109/ACCESS.2020.2991462
30. Chen C, Li K, Teo SG, Zou X, Wang K, Wang J, Zeng Z. Gated residual recurrent graph neural networks for traffic prediction. InProceedings of the AAAI conference on artificial intelligence 2019 Jul 17 (Vol. 33, No. 01, pp. 485-492). https://doi.org/10.1609/aaai.v33i01.3301485
31. Ge L, Li H, Liu J, Zhou A. Temporal graph convolutional networks for traffic speed prediction considering external factors. In2019 20th IEEE international conference on mobile data management (MDM) 2019 Jun 10 (pp. 234-242). IEEE. http://dx.doi.org/10.1109/MDM.2019.00-52
32. Xie Q, Guo T, Chen Y, Xiao Y, Wang X, Zhao BY. " how do urban incidents affect traffic speed?" A deep graph convolutional network for incident-driven traffic speed prediction. arXiv preprint arXiv:1912.01242. 2019 Dec 3. https://doi.org/10.48550/arXiv.1912.01242
33. Cao S, Wu L, Zhang R, Li J, Wu D. Capturing local and global spatial-temporal correlations of spatial-temporal graph data for traffic flow prediction. In2022 International Joint Conference on Neural Networks (IJCNN) 2022 Jul 18 (pp. 1-8). IEEE. http://dx.doi.org/10.1109/IJCNN55064.2022.9892616
34. Hu J, Lin X, Wang C. Dstgcn: Dynamic spatial-temporal graph convolutional network for traffic prediction. IEEE Sensors Journal. 2022 May 18;22(13):13116-24. https://doi.org/10.1109/JSEN.2022.3176016
35. Raschka S, Patterson J, Nolet C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information. 2020 Apr;11(4):193. https://doi.org/10.3390/info11040193
36. Zheng G. Deep Learning Models for Traffic Prediction in Urban Transport Networks: Bournemouth University; 2022.
37. Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine. 2013 Apr 5;30(3):83-98. https://doi.org/10.1109/MSP.2012.2235192
38. Rong Y, Huang W, Xu T, Huang J. Dropedge: Towards deep graph convolutional networks on node classification. arXiv preprint arXiv:1907.10903. 2019 Jul 25. https://doi.org/10.48550/arXiv.1907.10903
39. Li Y, Yu R, Shahabi C, Liu Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926. 2017 Jul 6. https://doi.org/10.48550/arXiv.1707.01926