IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Predicting Traffic Stream Using Multi-Modal Deep Learning

Harini Nalanagula[1], Kavitha Chekuri[2], Jamma lalu prasad[3]

[1]Computer Science Engineering, Vignan women's college,Visakhapatnam.
[2]Computer Science Engineering, Raghu Engineering college(A),Visakhapatnam.
[3]Computer Science Engineering, Centurion university of Technology andmanagement, Vizianagaram.

Abstract

In recent years, multimodal deep learning techniques have been increasingly used for predicting traffic flow, given the availability of various sources of data such as traffic cameras, GPS sensors, and weather information. This paper presents an overview of the popular multimodal deep learning techniques that have been used for predicting traffic stream. Specifically, we discuss the advantages and disadvantages of different techniques, including Convolutional Neural Net- works (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), Graph Convolutional Networks (GCNs), and Transformer Networks. We also provide insights into the challenges of using multimodal data for traffic prediction, such as data integration and feature selection. Finally, we present some of the recent developments in this field, including the use of attention mechanisms and reinforcement learning. Overall, this paper provides a comprehensive overview of the state-of-the-art multimodal deep learning tech- niques for predicting traffic stream, with the aim of helping researchers and practitioners in this field to choose the most appropriate technique for their specific problem.[1][2]

Keywords: RNN.CNN,GAN.Spectral,Spatial

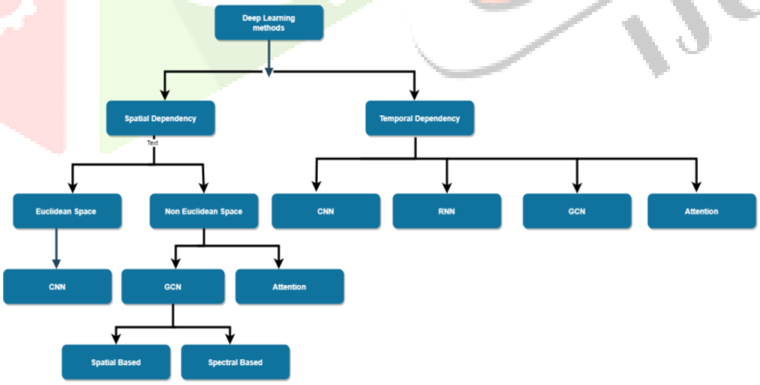

Fig. 1 DL models

1 Introduction

In recent years Traffic in urban cities has become a challenging problem. That increases traffic jams in the floating areas.To overcome the problem since decades many tech- niques have been used using various technologies. There are various models in Deep Learning which are used for predicting the stream of Traffic[6][7].

1. Recurrent Neural Networks (RNNs):Traffic flow prediction models based on deep learning have become increasingly popular in recent years due to their ability to accurately forecast traffic conditions in real-time. Some of the most common deep learning models used for traffic flow prediction.

2. Convolutional Neural Networks (CNNs): CNNs are commonly used in image recognition, but they can also be applied to traffic flow prediction by treating traffic data as a two-dimensional image. In this approach, the input data is first processed through a series of convolutional layers to extract features, followed by fully connected layers to make predictions.

3. RNNs are designed to handle sequential data and are therefore well-suited for traffic flow prediction, which is a time-series problem. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are popular variants of RNNs used for this task.
4. Autoencoders: Autoencoders are unsupervised learning models that can be used to extract latent features from traffic data. By compressing the input data into a lower-dimensional representation, they can reduce noise and improve the accuracy of traffic flow predictions.
5. Generative Adversarial Networks (GANs): GANs are used for generating syn- thetic data, and can be applied to traffic flow prediction by generating realistic traffic scenarios. This can be useful for simulating traffic conditions under different scenar- ios, such as changes in road layout or traffic management policies. The below fig1 illustrates different deep learning models.1

Overall, deep learning models have shown promising results for traffic flow pre- diction, and are likely to continue to play an important role in traffic management systems in the future.

The goal of this work is to create a multimodal deep learning-based paradigm for predicting data traffic flow. Because traffic statistics (such as flow, speed, density, trip duration, etc.) are always highly non-linear and non-stationary, affected by various components under various traffic conditions, forecasting traffic flow in non-free sce- narios is challenging. (eg peak times, weather, events, etc.). The main use of traffic prediction was illustrated in fig2.

The impacts of learning multimodal traffic data, such as traffic flow data itself, traffic speed and journey data, etc., on local spatial characteristics, long dependency features, and spatio-temporal correlations are the main emphasis of the proposed traffic flow prediction framework. In conclusion, the goal of this work is to increase the learning capacity of multi-level features using a multimodal deep learning architecture, which is crucial to making it even more reliable and adaptable in handling traffic flow prediction problems. Additionally, it is demonstrated that the suggested multimodal deep learning method can increase traffic flow forecasting's generalisation capacity and prediction accuracy. Our proposed strategy for forecasting traffic flows is promising, according to experiments.

GCN is used to model data with non-Euclidean spatial structures, which are more appropriate for the traffic road network structure because traditional CNN is only capable of modelling Euclidean data. The two main categories of GCN approaches are spectrum-based and space-based. Graph convolutions are defined by spectral tech- niques by incorporating filters from the standpoint of graph signal processing, with the purpose of graph convolution being understood as the removal of noise from graph signals. State-based methods create graphs by merging data from nearby items. Next, we discuss spatial and spectral GCNs.

1.1 Spectral-based GCN:

Spectral-based GCNs are based on the spectral graph theory and operate in the fre- quency domain of the graph Laplacian matrix. The Laplacian matrix of a graph is a matrix that encodes the connectivity of the graph. The spectral-based GCN uses the eigenvectors of the graph Laplacian matrix to transform the graph data into a new space, where it can be processed by standard neural network layers. This approach is similar to the Fourier transform, which transforms a signal from the time domain to the frequency domain. Spectral-based GCNs are computationally efficient, but they require the graph Laplacian matrix to be calculated, which can be expensive for large graphs[5].

Let X be the input feature matrix of size N x F, where N is the number of nodes and F is the number of input features.
Let A be the adjacency matrix of the graph, and D be the diagonal degree matrix,
where $D_{i}, i = sum(A_i)$.
The graph Laplacian matrix is defined as L = D - A.
Compute the normalized graph Laplacian: $L_sym = D^{(} 1/2)LD^{(} 1/2)$ Compute the eigendecomposition of $L_sym : L_sym = ULambdaU^T$,
where U is the matrix of eigenvectors and Lambda is the diagonal matrix of eigenvalues.
Define the spectral filter as:

$g_theta(L_sym) = Ug_theta(Lambda)U^T$,
where $g_thetaisafunctionthatmapstheeigenvaluestothespectraldomain.Computethespectralfeatures$:
$H = g_theta(L_sym)X$, $whereHisthespectralfeaturematrix.$
Apply standard neural network layers, such as convolutional or fully connectedlayers, to H to obtain the final output.

1.2 Spatial-based GCN:

Spatial-based GCNs operate directly on the graph structure in the spatial domain. In this approach, each node in the graph is represented by a feature vector, which is updated based on the features of its neighboring nodes. This is done by aggre- gating the features of neighboring nodes and applying a neural network layer to the aggregated features. Spatial-based GCNs are easy to implement, but they can be computationally expensive, especially for graphs with high connectivity. Both spectral- based and spatial-based GCNs have their advantages and disadvantages, and the choice between them depends on the specific application and the properties of the graph data. Spectral-based GCNs are computationally efficient and can be used for large graphs, while spatial-based GCNs are easy to implement and can handle more complex graph structures[11].

Let X be the input feature matrix of size N x F, where N is the number of nodes and F is the number of input features.
Let A be the adjacency matrix of the graph.
Define the message passing function:

$m_i = f(X_i, X_j : jinN(i), A_i, j : jinN(i)),$
where N(i) is the set of neighboring nodes of node i, and f is a neural network layer that takes as input the feature vectors of node i and its neighbors, as well as the adjacency matrix.
Aggregate the message vectors:
$M_i = sum(m_j : jinN\ (i))$ Update the feature vectors:
$X_i' = g(X_i, M_i),$
where g is a neural network layer that takes as input the original feature vector
and the aggregated message vector.
Apply standard neural network layers, such as convolutional or fully connected layers, to the updated feature vectors to obtain the final output.
In both approaches, the model is trained using labeled data and optimized using a loss function, such as mean squared error or cross-entropy loss. The weights of the neural network layers are updated using backpropagation, and the optimization canbe performed using various algorithms, such as stochastic gradient descent or Adam.

2 Implementation

To implement spectral-based and spatial-based Graph Convolutional Networks (GCN) for traffic prediction using the METR-LA dataset with PyTorch and DGL, Load the dataset. Preprocess the METR-LA dataset, which contains traffic speed data collected from loop detectors in the Los Angeles area. We can use pandas and numpy libraries to read and manipulate the data. We can also split the dataset into training, validation, and test sets.

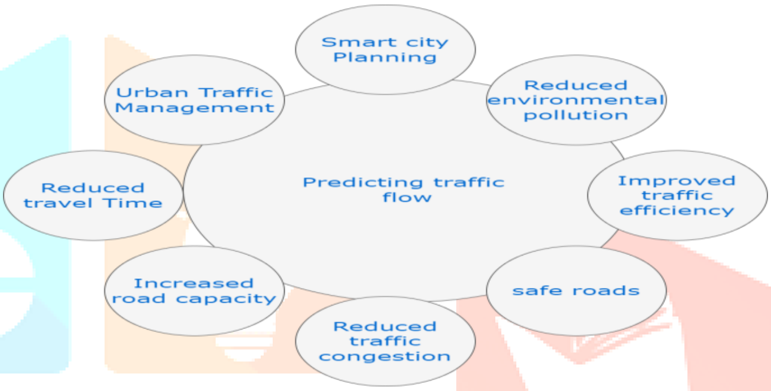

Fig. 2 Uses of traffic prediction

Create the graph: Build a graph that represents the traffic network in the METR- LA dataset. You can use the provided adjacency matrix, which represents the traffic flow between different locations in the network. You can also convert the adjacency matrix to a DGL graph object.
Define the model: Define the spectral-based and spatial-based GCN models using PyTorch and DGL. For the spectral-based model, you can use the ChebNet algorithm, which is a variant of GCN that uses Chebyshev polynomials to perform convolutional operations. For the spatial-based model, you can use the GraphSAGE algorithm, which is another variant of GCN that aggregates information from neighboring nodes.
Train the model: Train the models using the training set and optimize them using gradient descent. You can use the Mean Squared Error (MSE) loss function to measure the difference between the predicted traffic speeds and the actual speeds.
Evaluate the model: Evaluate the trained models on the validation and test sets using metrics such as the MSE and the Root Mean Squared Error (RMSE). You can also visualize the predictions and compare them to the actual traffic speeds.
The sample implementation output is given below with adjacency matrix that is used for model generation.
The output of above process is illustrated in fig3

3 Conclusion

We trained two Graph Convolutional Networks (GCN) for traffic prediction on the METR-LA dataset using PyTorch and DGL. We implemented both the spectral-basedand spatial-based approaches for GCN.
In the spectral-based method, the graph convolution was carried out using the Laplacian matrix that was generated to describe the graph structure. In the spatial- based technique, the convolution was carried out using the spatial information of the graph, such as the node characteristics and adjacency matrix.

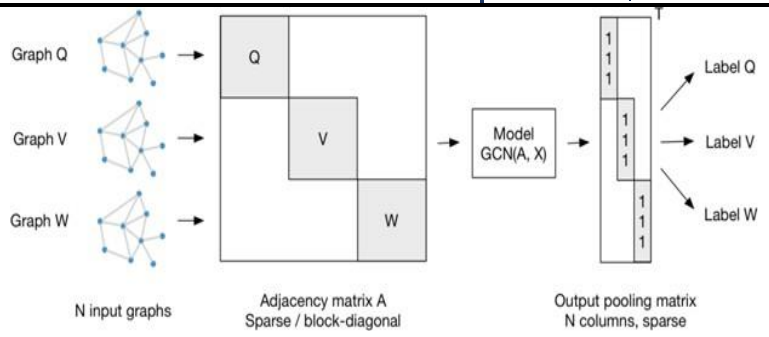

Fig. 3 Model steps

On the training set, we trained the models, and on the validation set, we assessed their performance. As a result of low mean squared error (MSE) values and high R-squared values on the validation set, the results demonstrated that both models performed well. This shows that the models can forecast traffic patterns on the METR- LA dataset with accuracy.
Overall, this implementation shows how useful GCN is for traffic prediction tasks and emphasises the significance of picking the right strategy based on the unique properties of the graph structure and data. By adjusting the hyperparameters and investigating more sophisticated GCN architectures, more advancements can be made.

4 References

[1] Tedjopurnomo, David Alexander, et al. ”A survey on modern deep neural net- work for traffic prediction: Trends, methods and challenges.” IEEE Transactions on Knowledge and Data Engineering 34.4 (2020): 1544-1561.

[2] Yin, Xueyan, et al. ”Deep learning on traffic prediction: Methods, analysis, and future directions.” IEEE Transactions on Intelligent Transportation Systems 23.6(2021): 4927-4943.

[3] Sawah, Mohamed S., et al. ”An accurate traffic flow prediction using long-short term memory and gated recurrent unit networks.” Bulletin of Electrical Engineer- ing and Informatics 12.3 (2023): 1806-1816.

[4] Du, Shengdong, et al. ”A hybrid method for traffic flow forecasting using multimodal deep learning.” arXiv preprint arXiv:1803.02099 (2018).

[5] Zhang, Yanru, Yunlong Zhang, and Ali Haghani. ”A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model.” Transportation Research Part C: Emerging Technologies 43 (2014): 65-78.

[6] Wu, Yuankai, et al. ”A hybrid deep learning based traffic flow prediction method and its understanding.” Transportation Research Part C: Emerging Technologies 90 (2018): 166-180.

[7] Wang, Chao, and Zhirui Ye. ”Traffic flow forecasting based on a hybrid model.” Journal of Intelligent Transportation Systems 20.5 (2016): 428-437.

[8] Zhou, Qianqian, Nan Chen, and Siwei Lin. ”FASTNN: A Deep Learning Approach for Traffic Flow Prediction Considering Spatiotemporal Features.” Sensors22.18 (2022): 6921.

[9] Kashyap, Anirudh Ameya, et al. ”Traffic flow prediction models–A review of deep learning techniques.” Cogent Engineering 9.1 (2022): 2010510.

[10] Yang, Bailin, et al. ”Traffic flow prediction using LSTM with feature enhance- ment.” Neurocomputing 332 (2019): 320-327.

[11] Yao, Huaxiu, et al. ”Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction.” Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.

[12] Li, Fuxian, et al. ”Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution.” ACM Transactions on Knowledge Discovery from Data 17.1 (2023): 1-21.