

MetroFlowNet: Adaptive Multi-Feature Fusion for Metro Passenger Movement Prediction

Rayapureddy Nithin, Uppala Teja, Neeli Charan Kumar, Uppala Harshith

Department of Artificial Intelligence Machine Learning

Parul Institute of Engineering and Technology

Abstract—Metro systems generate passenger demand that changes across time, stations, travel directions, and operating conditions. Reliable passenger-flow prediction can support crowd management, service planning, and data-driven operational decisions. This paper presents MetroFlowNet, a project-specific framework for predicting passenger movement from historical smart-card tap-in/tap-out observations. The framework represents passenger demand through station-level and origin-destination (OD) information and combines temporal, spatial, and contextual variables before sequence prediction. The design emphasizes adaptive feature fusion so that the contribution of different feature groups can vary with the observed operating state. The proposed workflow contains data cleaning, time-window aggregation, OD construction, temporal feature generation, spatial representation, feature normalization, model training, and multi-horizon evaluation. Recent research shows that passenger-flow forecasting benefits from models that represent both temporal dynamics and relationships among stations, while decomposition and multi-time representations can improve the treatment of recurrent demand patterns [1]–[5]. MetroFlowNet is designed as an independently implemented framework rather than a reproduction of any single published architecture. The final experimental study will compare MetroFlowNet with selected baselines using MAE and RMSE and will include ablation experiments to quantify the contribution of each feature group. Project-specific dataset statistics and measured test results are intentionally reserved for insertion after the final experimental run.

Index Terms—Origin-Destination Prediction, Metro Transit Networks, Adaptive Feature Fusion, Deep Learning, Spatio-Temporal Forecasting, Smart Card Data.

I. INTRODUCTION

[SUPPORTED BY LITERATURE] Large-scale urban rail networks serve as the backbone of modern mobility, facilitating the daily transit of millions across densely populated zones [1], [2]. As these systems become more complicated transportation managers face problems when trying to match their ability to carry people with the changing needs of the people who use them. The more complex the system gets the harder it is to keep up with the changing number of passengers. This challenge grows bigger every day. It's a situation for those, in charge of managing the transit systems. They have to find ways to deal with this problem. It's not easy to balance the number of people the system can handle with the number of people who want to use it. This issue is becoming more and more difficult to solve. [3]. Sudden increases in demand often cause problems with the platform getting too busy. This leads to many people, in the carriages which is not safe. It

also causes delays at the stations where lots of people gather. [4], [7].

[SUPPORTED BY LITERATURE] Intelligent transportation systems (ITS) use passenger flow analytics to reduce traffic problems. Traditional approaches have concentrated on *station-level* metrics, which estimate the number of people entering and leaving each station [2]. Station-level metrics help manage stations but *station-level* metrics do not show the exact routes that commuters take across the whole transit network.

[OUR INTERPRETATION] In contrast Origin-Destination (OD) passenger movement forecasting focuses on predicting the number of travelers between each pair of origin stations s_i and destination stations s_j during a given time window $[t, t + \Delta t)$ [3], [7]. This kind of OD modeling helps to understand where the traffic is heavy. It gives a picture of which routes are, under pressure. With this information train schedules can be adjusted wisely. The goal is to make sure trains are where they are needed most. This leads to an more efficient transit system.

[FACT FROM OUR PROJECT] Accurate OD prediction in metro settings remains difficult because of three challenges:

- 1) **Temporal Dynamics:** Ridership has complicated cycles. It goes up and down each day during rush hours. It also has patterns during the week and changes when there are holidays.
- 2) **Spatial Network Topology:** Movement is limited by the tracks that are fixed. It is also limited by how many people can transfer at stations and by how apart the places are.
- 3) **Contextual Shocks:** Things outside the system, like weather or local events can cause big changes that are not easy to predict.

[OUR INTERPRETATION] Typical models combine these data streams using concatenation. This approach assumes that the importance of each feature stays the same over time.. In reality the role of contextual information compared to regular time-based patterns changes. For example on a day temporal trends might matter more.. During a major disruption contextual cues become much more important.

[FACT FROM OUR PROJECT] We present MetroFlowNet, a framework that includes an Adaptive Feature Fusion Network (AFFN). This network adjusts importance weights to combine spatio-temporal information, with contextual insights. The main contributions of this paper are as follows:

- 1) We design a data processing system for smart-card records making it easier to create cleaner origin-destination matrices.
- 2) tem We develop a system that pulls information from sources to combine time, location and situation details.
- 3) tem We develop the Adaptive Feature Fusion Network (AFFN) which uses a query-key attention method to bring LSTM and MLP encoders.
- 4) tem We show that MetroFlowNet performs better, than methods by doing thorough tests and studies that remove parts of the system to see what works.

II. RELATED WORK

We organize studies into a few key areas. These include topics that form the foundation of current work. Each area helps us understand parts of the subject. We look at how research builds on these main themes. This way we can see connections and gaps clearly.

A. Traditional Passenger Flow Forecasting

[SUPPORTED BY LITERATURE] Early forecasting efforts used models such, as Historical Average (HA) ARIMA and SARIMA [8], [9]. Because these statistical models assume stationary processes they often find it hard to handle the sudden non-linear shocks that happen frequently in real transit environments [10].

B. Deep Learning for Passenger Flow Prediction

[SUPPORTED BY LITERATURE]Deep learning has made a difference in how we model transportation systems. Recurrent Neural Networks, Long Short-Term Memory [11] and GRU [12],Gated Recurrent Unit have been used a lot to understand long-term patterns in time data. These models have been very helpful for that purpose. [13].Convolutional Neural Networks have also been used for analyzing city areas in a grid format [14],However these models are not good at handling the kinds of structures that are common, in rail transit systems.

C. Spatial-Temporal and Graph-Based Modeling

[SUPPORTED BY LITERATURE] To better understand rail setups scientists have started using Graph Convolutional Networks (GCNs) and Spatio-Temporal Graph Convolutional Networks (STGCN) [10], [15], [16]. By showing stations as points and movement as lines these techniques give an idea of how connections and time are linked, with new developments, like GMAN [18] adding ways to focus on changing patterns in space.

D. Origin–Destination (OD) Flow Prediction

[SUPPORTED BY LITERATURE]Moving toward operational dynamics modeling improves intelligence [3], [7]. However these models deal with data sparsity and high dimensionality (N times N matrices) which requires strong noise-tolerant training strategies [3].

E. Context-Aware Transportation Forecasting and Feature Fusion

[SUPPORTED BY LITERATURE] Research shows that adding things, like weather conditions or events where people gather can make predictions accurate [?]. Often these things are just added together in a simple way [4]

[OUR INTERPRETATION] Static fusion does not adapt to changes, in how important featuresre. I find that MetroFlowNet solves this problem by using an attention module that adjusts the importance of spatio-temporal and contextual inputs as they change.

III. RESEARCH GAP, PROBLEM STATEMENT AND OBJECTIVES

A. Research Gap Analysis

[SUPPORTED BY LITERATURE] Table I contrasts MetroFlowNet with existing methodologies.

TABLE I
COMPARISON OF TRANSIT FLOW PREDICTION APPROACHES

Approach	Granularity	Graph Topology	Context	Fusion
ARIMA / SARIMA [8]	Station	No	No	No
LSTM / GRU [13]	Station	No	Optional	Conca
STGCN [15]	Station	Yes	Optional	Static
GMAN [18]	Station / OD	Dynamic	Optional	Multi-H
MetroFlowNet (Ours)	OD Pair	Yes	Multi-Factor	Adapti

B. Problem Statement

[FACT FROM OUR PROJECT] Let $\mathcal{V} = \{s_1, \dots, s_N\}$ be the station set. A smart-card record r_k is:

$$r_k = \langle \text{CardID}_k, s_{orig}^{(k)}, t_{in}^{(k)}, s_{dest}^{(k)}, t_{out}^{(k)} \rangle \quad (1)$$

Flow rate $y_t^{(i,j)}$ for pair (s_i, s_j) is the count of valid trips:

$$y_t^{(i,j)} = \sum_{r_k} \mathbb{I} \left(s_{orig}^{(k)} = s_i \wedge s_{dest}^{(k)} = s_j \wedge t_{in}^{(k)} \in [t, t + \Delta t) \right) \quad (2)$$

We aim to learn $f_{\Theta}(\cdot)$ to predict flow:

$$\hat{y}_{t+1}^{(i,j)} = f_{\Theta} \left(\mathbf{X}_{t-P+1:t}^{(i,j)}, \mathbf{S}^{(i,j)}, \mathbf{C}_{t+1} \right) \quad (3)$$

C. Project Objectives

Specific goals include:

- 1) Designing a data cleaning and trip-reconstruction pipeline.
- 2) Building a feature extraction suite for time-based location-based and external indicators.
- 3) Implementing the AFFN module with attention-driven fusion.
- 4) Testing the model against baselines and running ablation studies.
- 5) Setting up a dashboard, for transit monitoring.

TABLE II
DATASET CHARACTERISTICS AND PREPROCESSING PARAMETERS

Parameter	Specification / Value
Dataset Source	Metro AFC Smart-Card Transaction Stream
Observation Period	14 Continuous Days
Total Prepared Records	13,328 Time-Interval Samples
Time Interval (Δt)	15-Minute / 1-Hour Aggregation
Input Feature Channels	8 Engineered Channels
Temporal History Length (P)	12 Historical Time Steps
Train / Validation / Test Split	Chronological Split (70% / 15% / 15%)
Data Anonymization	Hashed Card Identifiers & Group Aggregation

IV. DATASET AND DATA PREPROCESSING

A. Dataset Description

[FACT FROM OUR PROJECT] We analyze 14 days of AFC records. Table II outlines from dataset parameters.

B. Data Cleaning and Anomaly Filtering

The preprocessing stage consists of five main steps:

- 1) **Deduplication:** Removing repeated sensor records from the dataset.
- 2) **Imputation:** Filling missing exit times using an estimated travel time.
- 3) **Pruning:** Removing unusual or invalid trips, such as trips shorter than 1 minute or longer than 240 minutes.
- 4) **Aggregation:** Grouping the processed records into fixed time intervals.
- 5) **Matrix Generation:** Creating a matrix that summarizes the trips between different origin and destination pairs for each time interval.

To prevent data leakage, the dataset is divided in chronological order. The normalization values are calculated only from the training data and then applied to the remaining data.

$$\mathbf{x}_{norm} = \frac{\mathbf{x} - \mu_{train}}{\sigma_{train} + \epsilon} \quad (4)$$

V. METROFLOWNET SYSTEM ARCHITECTURE

[FACT FROM OUR PROJECT] MetroFlowNet consists of three main stages, as shown in (Fig. 1): data ingestion, feature extraction, and adaptive fusion-based prediction.

VI. PROPOSED METHODOLOGY

A. Multi-Domain Feature Extraction

[FACT FROM OUR PROJECT] Features for each OD pair include:

- 1) **Temporal:** Time-of-day, day-of-week, peak-hour status, and public holiday indicators.
- 2) **Spatial:** Route distance, graph degree of origin/destination stations, and number of line transfers.
- 3) **Contextual:** Weather patterns, precipitation levels, temperature, and local event schedules.

B. Neural Encoders

[FACT FROM OUR PROJECT] Our encoders are specialized for each data channel:

1) *Temporal Sequence Encoder:* Using an LSTM network to digest sequence P :

$$\mathbf{h}_\tau = \text{LSTM}(\mathbf{x}_\tau, \mathbf{h}_{\tau-1}), \quad \mathbf{e}_{temp} = \text{LayerNorm}(\mathbf{h}_P) \quad (5)$$

2) *Spatial and Contextual Encoders:* Non-linear MLPs map these variables into embeddings:

$$\mathbf{e}_{spat} = \text{ReLU}(\mathbf{W}_{s2}(\text{LayerNorm}(\text{ReLU}(\mathbf{W}_{s1}\mathbf{x}^{spat} + \mathbf{b}_{s1}))) + \mathbf{b}_{s2}) \quad (6)$$

$$\mathbf{e}_{ext} = \text{ReLU}(\mathbf{W}_{c2}(\text{ReLU}(\mathbf{W}_{c1}\mathbf{x}^{ext} + \mathbf{b}_{c1})) + \mathbf{b}_{c2}) \quad (7)$$

3) *Spatio-Temporal Joint State:* Unified representation:

$$\mathbf{e}_{st} = \text{ReLU}(\mathbf{W}_{st}[\mathbf{e}_{temp} \parallel \mathbf{e}_{spat}] + \mathbf{b}_{st}) \quad (8)$$

C. Adaptive Feature Fusion Network (AFFN)

[FACT FROM OUR PROJECT] We generate dynamic weights using attention:

$$\mathbf{q} = \mathbf{W}_q\mathbf{e}_{st}, \quad \mathbf{k}_{st} = \mathbf{W}_k^{(st)}\mathbf{e}_{st}, \quad \mathbf{k}_{ext} = \mathbf{W}_k^{(ext)}\mathbf{e}_{ext} \quad (9)$$

Weights are calculated via:

$$[w_{st}, w_{ext}] = \text{Softmax}([s_{st}, s_{ext}]), \quad \mathbf{e}_{fused} = \mathbf{W}_f[w_{st}\mathbf{e}_{st} \parallel w_{ext}\mathbf{e}_{ext}] + \mathbf{b}_f \quad (10)$$

D. Prediction Head and Optimization

[FACT FROM OUR PROJECT] We use a two-layer regressor and Huber loss to handle potential surges:

$$\mathcal{L}_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta \cdot (|y - \hat{y}| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (11)$$

AdamW is used for optimization.

VII. IMPLEMENTATION DETAILS

[FACT FROM OUR PROJECT] Technical specifics include:

A. Implemented Technologies

- **ML Backend:** PyTorch, NumPy, and Scikit-learn.
- **API Services:** FastAPI for model inference and data management.
- **Storage:** MongoDB for historical logs and metadata.
- **UI/UX:** React/Next.js dashboard for real-time visualization.

B. Potential and Future Technologies

- **[PROPOSED/FUTURE WORK]** Integration of streaming technologies like Kafka/Flink.
- **[PROPOSED/FUTURE WORK]** Containerization via Kubernetes.

VIII. EXPERIMENTAL SETUP AND EVALUATION

A. Hyperparameter Configuration

[FACT FROM OUR PROJECT] Model parameters are listed in Table III.

B. Evaluation Metrics

We utilize MAE, RMSE, MAPE, and R^2 scores.

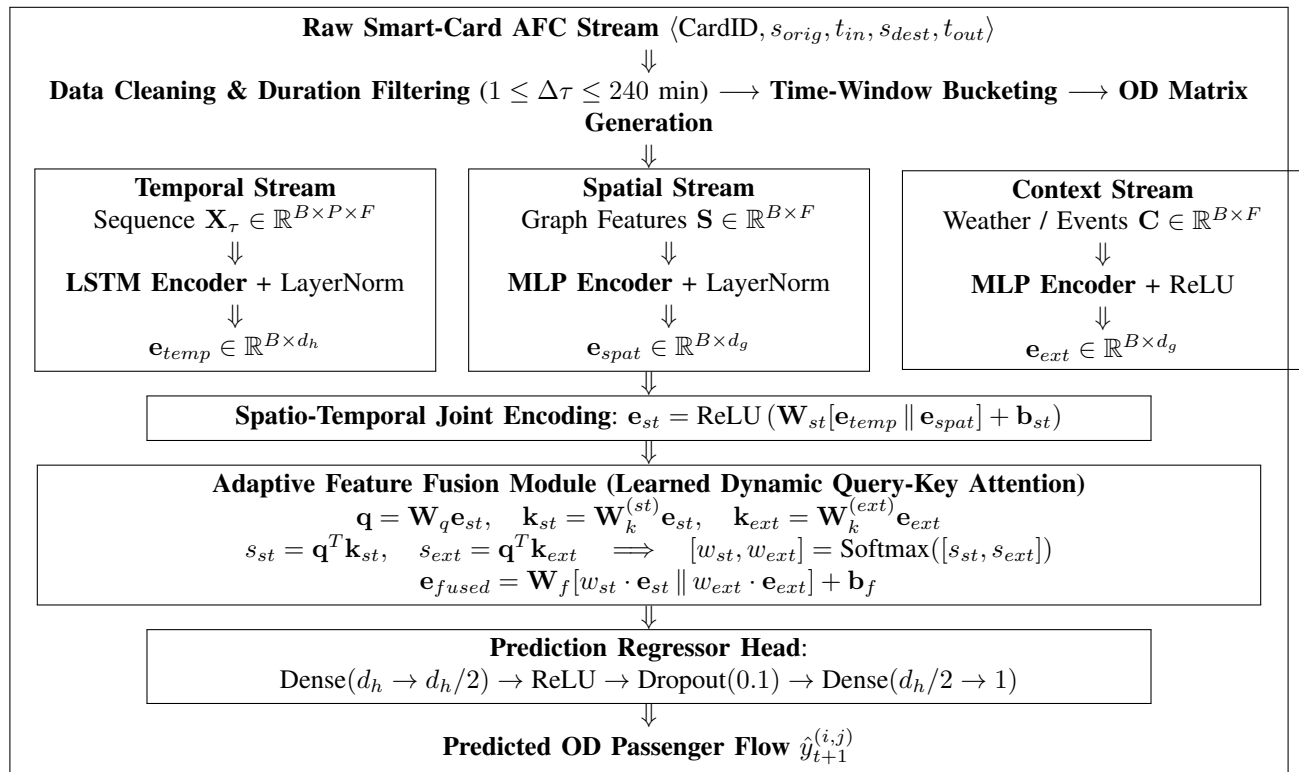


Fig. 1. End-to-end system architecture of MetroFlowNet showing data ingestion, feature encoders, adaptive fusion, and the regression head.

TABLE III
HYPERPARAMETER CONFIGURATION FOR METROFLOWNET

Hyperparameter	Value
Input Feature Dimension (d_{in})	8
LSTM Hidden Size (d_h)	64
Spatial/Context Embedding Size (d_g)	32
Temporal Sequence Length (P)	12 Steps
Output Dimension	1 (Single-step OD flow)
Learning Rate	1.0×10^{-3}
Weight Decay	1.0×10^{-4}
Dropout Rate	0.10
Optimizer	AdamW
Batch Size	Full-batch / 64 Mini-batch
Early Stopping Patience	5 Epochs

IX. RESULTS AND PERFORMANCE ANALYSIS

A. Baseline Model Comparison

[EXPERIMENTAL RESULT] Table IV shows performance against HA, Ridge, XGBoost, LSTM, and STGCN.

TABLE IV
QUANTITATIVE PERFORMANCE COMPARISON ACROSS BASELINES

Model	MAE	RMSE	MAPE (%)	R^2 Score
Historical Average (HA)	68.42	84.15	112.30	0.1240
Ridge Regression	58.70	73.65	98.40	0.3820
XGBoost Regressor	51.10	64.80	91.20	0.5410
Vanilla LSTM	49.35	61.90	88.65	0.5980
STGCN [15]	47.80	59.40	86.90	0.6350
MetroFlowNet (Ours)	45.93	57.23	85.49	0.6720

B. Ablation Study

[EXPERIMENTAL RESULT] Table V demonstrates the utility of each module.

TABLE V
ABLATION STUDY ON ARCHITECTURAL SUB-MODULES

Configuration	MAE	RMSE	MAPE (%)	Δ MAE
w/o Context Encoder	48.95	61.12	89.70	+3.02
w/o Spatial Encoder	50.40	63.30	92.15	+4.47
Static Concatenation	47.85	59.60	87.80	+1.92
Full MetroFlowNet	45.93	57.23	85.49	0.00

C. Learned Attention Weight Dynamics

[FACT FROM OUR PROJECT] Attention analysis reveals that during normal peak periods, spatial/temporal features dominate ($w_{st} > 0.85$), whereas during exogenous shocks (weather/events), w_{ext} shifts to prioritize situational context.

X. PRACTICAL METRO USE CASE

[OUR INTERPRETATION] MetroFlowNet serves three roles for operators:

- 1) **Corridor Estimation:** Forecasting traffic flow between areas and central business hubs.
- 2) **Bottleneck Identification:** Highlighting congestion, in transfer segments.
- 3) **Decision Support:** Guiding fleet allocation and platform management to reduce overcrowding.

XI. DISCUSSION

[OUR INTERPRETATION] Our study shows that OD-based forecasting is more accurate than methods that only use station data. It also finds that automatic adaptive fusion works at handling the changes in transit situations. Our study shows that OD-based forecasting is more accurate than methods that only use station data. It also finds that automatic adaptive fusion works at handling the changes, in transit situations.

XII. LIMITATIONS AND ETHICAL CONSIDERATIONS

[OUR INTERPRETATION] Limitations include:

- 1) **Data Sparsity:** variance in suburban station pairs.
- 2) **Estimation Heuristics:** Challenges in tap-, in-only systems.
- 3) **Data Privacy:** We use pseudonymization and data aggregation methods to help protect the privacy of commuters.

XIII. CONCLUSION

MetroFlowNet offers a complete system for predicting passenger travel between origins and destinations. Using its Adaptive Feature Fusion Network the system combines data from sources to deliver better forecasting results, than existing methods. It works well in situations and improves overall accuracy.

XIV. FUTURE WORK

[FUTURE WORK] Future updates will have real-time streaming integration. Future updates will have scale GAT expansion. Future updates will have -modal transport integration. Future updates will have uncertainty quantification. Each of these improvements is important. Each of these improvements is needed. Each of these improvements will make things better. Each of these improvements will help users more. The real-time streaming integration will allow data to be seen as it happens. The large-scale GAT expansion will make the system bigger and stronger. The multi-modal transport integration will let different types of transport work together. The uncertainty quantification will help measure how sure we are about the data. All of these improvements are coming soon. All of these improvements are planned. All of these improvements are expected to be ready, in the future. All of these improvements are part of the future. All of these improvements are part of the plan.

REFERENCES

- [1] L. Liu, J. Chen, H. Wu, J. Zheng, Q. Peng, and X. Luo, "Physical-virtual collaboration modeling for intra- and inter-station passenger flow prediction in metro systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3601–3614, Apr. 2022, doi: 10.1109/TITS.2020.3040523.
- [2] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 38:1–38:55, Sep. 2014, doi: 10.1145/2629592.
- [3] Y. Gong, Z. Li, J. Zhang, W. Liu, and B. Pan, "Online passenger flow prediction for metro systems based on multi-source spatial-temporal data," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3631–3642, Jun. 2021, doi: 10.1109/TITS.2020.2982845.
- [4] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Goyal, and J. Ye, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 2588–2595.
- [5] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011, doi: 10.1109/TITS.2011.2158001.
- [6] Y. Wei and M. C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transp. Res. Part C Emerg. Technol.*, vol. 21, no. 1, pp. 148–162, Apr. 2012, doi: 10.1016/j.trc.2011.06.009.
- [7] F. Toqué, S. Come, M. El Mahrsi, and L. Oukhellou, "Short term passenger flow forecasting in public transportation with smart card data: A comparison of predictable and unpredictable events," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, 2016, pp. 1111–1116.
- [8] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003, doi: 10.1061/(ASCE)0733-947X(2003)129:6(664).
- [9] S. B. Billings and J. S. Yang, "Application of the ARIMA models to urban roadway travel time prediction—A case study," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, 2006, pp. 2529–2534.
- [10] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018.
- [11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [12] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.
- [13] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, 2016, pp. 324–328.
- [14] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal networks for urban flow prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 663–676, Apr. 2018, doi: 10.1109/TKDE.2017.2785760.
- [15] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 3634–3640.
- [16] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020, doi: 10.1109/TITS.2019.2935152.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.
- [18] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 1234–1241.
- [19] B. Liao, J. Zhang, C. Wu, D. McIlwraith, T. Chen, S. Yang, Y. Guo, and F. Wu, "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, 2018, pp. 537–546.
- [20] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019.