RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2668

Dynamic Fusion of Graph-Based Spatial and State-Space Temporal Features for Multi-Horizon Traffic Speed Forecasting

D. Sivabalaselvamani [a,*], D. Thiyagarajan [b], M. Menagadevi [c], S. Yasotha [d], K. Nanthini [e], Sankar Ganesh S [f]

[a] Department of Computer Applications, Kongu Engineering College, Erode, Tamil Nadu, India.
[b] Department of Computer Science and Engineering, Chennai Institute of Technology, Chennai, Tamil Nadu, India.
[c] Department of Computing Technologies, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur – 603203, Tamil Nadu, India.
[d] Department of Computer Science and Engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India.
[e] Department of Computer Science and Applications, Christ Academy Institute for Advanced Studies, Bengaluru, Karnataka, India.
[f] Department of Artificial Intelligence and Data Science, KPR Institute of Engineering and Technology, Arasur, Coimbatore 641407, Tamil Nadu, India.
* Corresponding Author Email: sivabalaselvamani@gmail.com

DOI: https://doi.org/10.54392/irjmt2668

Received: 25-07-2026; Revised: 02-11-2026; Accepted: 16-11-2026; Published: 26-11-2026

Abstract: Reliable traffic speed forecasting in Internet of Things (IoT)-enabled intelligent transportation systems requires effective modelling of spatial dependencies and evolving temporal patterns across traffic conditions. This study proposes GMDF-Net (Graph-Mamba Dynamic Fusion Network), a multi-horizon forecasting architecture that integrates complementary spatial and temporal representations. GMDF-Net combines an adaptive-graph spatial branch, which uses a fixed road-distance graph and a learned adaptive adjacency matrix, with a Mamba-inspired selective state-space temporal branch. An input-dependent, node- and time-aware softmax gate fuses the two representations. The model was evaluated on PeMSD7(M), comprising 228 sensors and five-minute traffic-speed observations, using a 12-step input window and forecast horizons of 15, 30 and 60 minutes. GMDF-Net achieved mean absolute errors (MAEs) of 2.29, 3.15 and 4.27 mph, root mean square errors (RMSEs) of 4.33, 6.10 and 8.11 mph, and mean absolute percentage errors (MAPEs) of 5.45%, 7.89% and 11.48% at the respective horizons. Its MAE was lower than the reported LSTM, GCN and STGCN results at all horizons and lower than the reported STGNN-FAM result at 15 and 30 minutes; the 60-minute MAE was 3.64% higher than STGNN-FAM. The reported DNNF-ITS results were lower across all horizons. Error analysis showed a right-skewed distribution, larger errors at lower-speed sensors and delayed forecasts around abrupt speed transitions. GMDF-Net provides a competitive adaptive-graph and state-space framework for multi-horizon traffic speed prediction. The findings characterize its predictive performance and benchmark behaviour and motivate further component-wise and repeated evaluation to quantify the contributions of its spatial, temporal and fusion components.

Keywords: Traffic Speed Forecasting, Spatiotemporal Learning, Adaptive Graph Neural Networks, State-Space Models, Mamba, Dynamic Feature Fusion.

1. Introduction

Intelligent Transportation Systems (ITS) have been greatly improved by the new technologies that have emerged in the Internet of Things (IoT), connected sensing infrastructure, and Artificial Intelligence (AI). Modern ITS combine sensing, communication, perception, prediction and decision-making technologies to enable efficient and intelligent transportation management. These advancements have made it possible to gather vast quantities of traffic data from a multitude of sensors and monitoring sites, thereby offering the potential for data-driven traffic speed forecasting and intelligent transportation management. In this setting, traffic speed forecasting is a key component in converting the information captured by the sensors into predictions that can be used for intelligent transportation operations.

Traffic speed forecasting is an essential component of ITS because reliable forecasts can support congestion management, route planning, traffic control, travel-time estimation, and efficient utilization of transportation infrastructure. Dynamic neural network

architectures have also been studied to improve the adaptability of neural representations under changing input conditions [1]. However, traffic conditions are inherently dynamic and non-stationary, and the relationships between historical observations and future traffic states may vary across different time periods, locations, and traffic regimes. IoT-enabled ITS provides the sensing and data-acquisition infrastructure through which distributed traffic sensors continuously generate and transmit traffic observations for intelligent transportation applications. Within this conceptual architecture, traffic prediction models operate as an analytics component that transforms sensor-generated observations into traffic speed forecasts. In the present study, the PeMSD7 (M) dataset is used as a representative sensor-network dataset for evaluating this predictive analytics component. Therefore, the IoT-enabled aspect of the proposed framework refers to the sensor-based data acquisition and transportation intelligence context, while the experimental contribution focuses on adaptive traffic speed forecasting rather than the implementation of IoT communication protocols, sensing hardware, or edge devices. This distinction positions GMDF-Net as a predictive analytics component within an IoT-enabled ITS context while keeping the experimental scope consistent with the available benchmark dataset.

A major research direction in traffic forecasting is the joint modelling of spatial and temporal dependencies. Traffic conditions at one location are often influenced by conditions at neighbouring or connected locations, while traffic states also evolve continuously over time. Graph-based neural networks provide a natural representation of transportation networks by modelling monitoring locations as graph nodes and their relationships as graph edges. Research on spatial-temporal graph data has demonstrated the importance of modelling dynamic and heterogeneous relationships for traffic forecasting [2]. Spatial-temporal graph neural networks have also emerged as an important approach for jointly learning spatial and temporal traffic dependencies [3].

Despite the effectiveness of graph-based models, transportation networks cannot always be represented adequately using a fixed spatial structure. Traffic relationships may change according to traffic volume, congestion level, time of day, and other dynamic conditions. Dynamic graph convolutional recurrent architectures have therefore been investigated to capture changing traffic relationships [4]. Dynamic adjacency matrices have been employed to represent time-varying spatial dependencies in traffic forecasting [5]. Generic dynamic graph convolutional approaches have also been developed to model evolving spatial relationships in traffic-flow prediction [6]. Dynamic spatial-aware graph transformers have further been explored for spatiotemporal traffic-flow forecasting [7]. Dynamic spatial-temporal graph convolutional recurrent architectures have been proposed to capture changing traffic dependencies [8]. Adaptive dynamic graph convolutional approaches have also been investigated for multi-step traffic forecasting [9].

Although dynamic graph approaches improve the modelling of changing spatial relationships, traffic prediction involves more than adapting the underlying graph structure. Different neural components may capture complementary characteristics of traffic behaviour. Spatial representations can describe relationships among traffic locations, temporal representations can characterize sequential traffic evolution, and, as in our earlier DNNF-ITS framework, contextual representations can capture recurring temporal or calendar-related patterns. The predictive usefulness of these representations may therefore vary across traffic regimes, motivating a fusion mechanism that can adapt their relative contributions.

Attention and fusion mechanisms provide alternative approaches for integrating complementary information. Graph attention-based spatial-temporal learning has been investigated for traffic-flow forecasting [10], while attention mechanisms have also been incorporated into spatiotemporal graph networks [11]. Dynamic graph convolution combined with attention fusion has further been explored [12]. Feature-fusion approaches have demonstrated the value of combining learned representations for traffic-speed prediction [13], while dynamic fusion has also been considered in transportation-related prediction tasks such as traffic-accident forecasting [14]. The present study is specifically motivated by the need for adaptive integration of independent spatial, temporal and contextual predictive components in a unified traffic speed forecasting framework, which is a key challenge for these studies.

Two further developments are relevant to the present study. First, adaptive graph learning, introduced for traffic forecasting by Graph WaveNet [15], learns a self-adaptive dependency matrix from node embeddings so that spatial relations that are absent from the predefined sensor graph can still be captured. Second, structured and selective state-space models (SSMs) [16, 17] have emerged as linear-time alternatives to recurrent and attention-based sequence models and have recently been extended to spatial-temporal graph learning [18]. These developments suggest that both the spatial and the temporal components of a fusion-based forecaster can be strengthened, while leaving open the question of how the resulting representations should be weighted under different traffic conditions.

Taken together, these studies indicate a need for further investigation of adaptive fusion at the level of complementary predictive representations in traffic speed forecasting. Existing studies have extensively investigated spatiotemporal representation learning, dynamic graph construction, adaptive graph learning,

attention mechanisms, and feature-level fusion. However, comparatively less attention has been given to dynamically adapting the relative contribution of multiple complementary neural predictive representations according to the characteristics of the observed traffic state. A fixed fusion strategy assigns unchanged importance to different predictive components even when their predictive usefulness may vary across traffic regimes.

1.1 Aim and Objectives

This study addresses this gap by proposing GMDF-Net (Graph-Mamba Dynamic Fusion Network), a two-branch neural architecture that combines an adaptive-graph spatial block with a Mamba-inspired state-space temporal block through an input-dependent dynamic fusion mechanism for multi-horizon traffic speed forecasting. GMDF-Net is developed from our earlier DNNF-ITS framework (Dynamic Neural Network Fusion for Intelligent Transportation Systems), which fused GCN-based spatial, Bi-GRU temporal and MLP-based contextual representations. In GMDF-Net the spatial and temporal extractors are replaced, the separate contextual branch is not used, and the dynamic fusion principle is retained. The model is designed to adapt the relative contribution of the two representations according to the observed traffic conditions.

The primary research question addressed in this study is:

Can an input-dependent gate that weights an adaptive-graph spatial representation and a state-space temporal representation improve multi-horizon traffic-speed prediction compared with the individual representations and with fixed fusion, and under which traffic conditions does it do so?

For the predecessor DNNF-ITS this question was examined through component, fusion-strategy and traffic-regime analyses, which are summarised in Section 3.7. For GMDF-Net it is addressed here through a benchmark comparison and a forecast-error analysis; confirming the contribution of its individual components is the main remaining experimental task (Section 4).

Accordingly, the objectives of this study are to: (i) develop the GMDF-Net framework for traffic speed forecasting; (ii) learn complementary spatial representations through fixed and adaptive adjacency and temporal representations through a selective state-space module; (iii) design an input-dependent fusion mechanism for dynamically adapting their contributions; (iv) compare the proposed framework with conventional baselines and with the predecessor DNNF-ITS at the 15-, 30- and 60-minute horizons; and (v) characterise the training behaviour, the growth and distribution of error with the forecast horizon, the variation of error across sensors and the qualitative forecast behaviour of the model. Objectives (i) to (iii) are addressed by the architecture of Sections 2.5 to 2.7 and the design comparison of Table 2, objective (iv) by Section 3.1 and Table 7, and objective (v) by Sections 3.2 to 3.6. The component-level validation of objective (iii) for GMDF-Net is not part of the present study (Section 4).

The main contributions of this study are as follows:

1. A dynamic fusion framework, GMDF-Net, that integrates an adaptive-graph spatial representation and a Mamba-inspired state-space temporal representation for multi-horizon traffic speed forecasting.
2. An input-dependent two-branch gate, formulated as a dense soft mixture of two experts with node- and time-dependent weights (Section 2.7), together with a comparison of its design with attention, concatenation and mixture-of-experts fusion (Table 2).
3. A benchmark comparison of GMDF-Net with conventional deep learning baselines, the strongest attention-based baseline and the predecessor DNNF-ITS on PeMSD7(M) at three forecast horizons, reported together with the cases in which GMDF-Net does not outperform them.
4. An analysis of training convergence, error growth with the forecast horizon, the error distribution and its tail, sensor-wise error variation, and forecast behaviour at abrupt speed transitions, including an identification of the main weakness of the current model.

Relationship to the predecessor model. GMDF-Net follows an earlier in-house three-branch model, DNNF-ITS, which has not been published and is therefore not cited. The present study reuses from that model the dataset and preprocessing pipeline (Sections 2.1 and 2.2), a short description of the contextual feature encoding (Section 2.3, although GMDF-Net does not use it), the general training settings (Table 5), and the baseline and DNNF-ITS results in Table 7, which are values obtained earlier. Section 3.7 summarises, without reproducing its tables, the fusion evidence obtained with DNNF-ITS. The architecture of GMDF-Net (Sections 2.5 to 2.7), the design comparison of Table 2, the GMDF-Net results and all analyses in Sections 3.1 to 3.6 are specific to this study.

1.2 Related Work

Existing studies and foundational methods are reviewed in Sections 1.2.1 to 1.2.6 and summarised in Table 1; Section 1.2.7 states the research gap addressed by this work.

1.2.1 Deep Learning for Intelligent Transportation and Traffic Forecasting

The combination of artificial intelligence and ITS has led to the use of deep learning in traffic perception, prediction, and decision-making applications. Deep learning has become an important approach for traffic forecasting because of its ability to learn nonlinear relationships from historical traffic observations. Comprehensive research has examined deep learning algorithms for traffic forecasting and compared their characteristics with classical forecasting approaches [19]. Dynamic neural network research has investigated adaptable neural representations for changing input conditions [1], and gated mixture-of-experts layers combine expert sub-networks through a learned input-dependent gate [20]. Different neural architectures have subsequently been investigated for traffic prediction. Dynamic transition convolutional neural networks have been developed for traffic-demand prediction [21]. Deep feature fusion models have also been investigated for traffic-speed prediction by combining learned traffic representations [13]. These studies demonstrate the effectiveness of neural architectures for learning complex traffic patterns; however, variations in traffic conditions may cause the predictive usefulness of different representations to change across traffic regimes.

1.2.2 Spatiotemporal Traffic Prediction

Traffic forecasting requires the simultaneous modelling of spatial dependencies among locations and temporal dependencies across successive observations. Graph neural networks provide a structured mechanism for representing transportation networks by treating traffic monitoring locations as graph nodes and their relationships as graph edges. Research on spatial-temporal graph data has demonstrated the importance of learning dynamic and heterogeneous traffic relationships [2].

Spatial-temporal graph neural networks have also been extensively investigated for traffic forecasting and identified as an important approach for learning complex traffic dependencies [3]. Adaptive and recurrent graph-based architectures have further enhanced this direction. Adaptive graph convolutional recurrent networks have been developed for traffic forecasting [22]. Temporal multi-graph convolutional networks have also been proposed to represent multiple spatial and temporal relationships for traffic-flow prediction [23]. These studies demonstrate the effectiveness of graph-based architectures for jointly learning spatial and temporal traffic dependencies.

1.2.3 Dynamic Graph Learning for Traffic Prediction

A fixed graph structure may not adequately represent traffic relationships because the interactions between traffic locations can change over time. Dynamic graph learning addresses this limitation by adapting spatial relationships according to changing traffic conditions. Dynamic graph convolutional recurrent networks have been developed to capture changing traffic relationships [4]. Dynamic adjacency matrices have been introduced to represent time-varying spatial dependencies in traffic forecasting [5]. Generic dynamic graph convolutional networks have also been proposed to model evolving spatial relationships for traffic-flow forecasting [6]. Transformer-based and adaptive graph architectures have further extended dynamic graph learning. Dynamic spatial-aware graph transformers have been investigated for spatiotemporal traffic-flow forecasting [7]. Dynamic spatial-temporal graph convolutional recurrent networks have been proposed to capture changing traffic dependencies [8]. Dynamic adaptive spatiotemporal graph convolutional networks have also been developed for multi-step traffic forecasting [9]. These studies demonstrate that dynamically adapting the graph structure can improve the representation of changing spatial dependencies. These studies mainly adapt the graph structure itself.

1.2.4 Attention and Fusion-Based Traffic Prediction

Attention mechanisms have been increasingly incorporated into traffic forecasting models to identify informative spatial, temporal, or feature-level relationships. Graph attention-based spatial-temporal learning has been investigated for traffic-flow forecasting [10]. Attention mechanisms have also been integrated into spatiotemporal graph networks for traffic-flow prediction [11]. Dynamic graph convolution combined with attention fusion has further been explored for traffic-flow prediction [12]. Fusion-based approaches provide another mechanism for exploiting complementary information. Deep feature fusion has been applied to traffic-speed prediction by combining learned traffic features [13]. Dynamic fusion has also been investigated for traffic-accident forecasting, demonstrating the potential of dynamically combining information in transportation prediction tasks [14]. These studies mainly perform feature fusion or attention-based weighting within one architecture.

1.2.5 Traffic Data Quality and Forecasting Reliability

The quality and completeness of traffic observations directly influence the reliability of data-driven forecasting. Traffic sensing systems can produce incomplete observations because of sensor failures, communication interruptions, or other operational conditions. Comprehensive research on traffic missing-data imputation has examined approaches for recovering incomplete traffic observations and improving the reliability of traffic data [24]. Missing-data handling is therefore an important consideration in practical traffic forecasting. Imputation concerns the quality of the inputs and is treated here as a preprocessing step (Section 2.1).

Table 1. Comparative summary of the reviewed studies and foundational methods cited in this work

Ref.	Study	Method	Application	Limitation / gap addressed by GMDF-Net
[1]	Han *et al.* (2021)	Survey of dynamic neural networks that adapt structure or parameters to each input	General deep learning	Not specific to traffic data; motivates input-dependent computation but does not address fusion of spatial and temporal traffic representations.
[2]	Guo *et al.* (2021)	Spatial-temporal graph learning	Traffic forecasting	Focuses on dynamic and heterogeneous graph relationships, not on fusing independent predictive components.
[3]	Bui *et al.* (2022)	Overview of spatial-temporal graph neural networks and open research issues	Traffic forecasting	Summarises models and open issues; does not propose or evaluate an adaptive fusion of separately learned representations.
[4]	Li *et al.* (2023)	Dynamic graph convolutional recurrent network	Traffic prediction	Adapts the graph structure to changing traffic relationships.
[5]	Shi *et al.* (2023)	Graph convolutional recurrent network with dynamic adjacency matrix (DAGCRN)	Traffic forecasting	Models time-varying spatial dependencies; graph-centred.
[6]	Xu *et al.* (2023)	Generic dynamic graph convolutional network	Traffic-flow forecasting	Models evolving spatial relationships; graph-centred.
[7]	Li *et al.* (2024)	Dynamic spatial-aware graph transformer	Spatiotemporal traffic-flow forecasting	Dynamic spatial modelling; relative contribution of components not adapted.
[8]	Xia *et al.* (2024)	Dynamic spatial-temporal graph convolutional recurrent network	Traffic-flow forecasting	Captures changing dependencies through the graph.
[9]	Cui *et al.* (2023)	Dynamic adaptive deeper spatio-temporal GCN (ADSTGCN)	Multi-step traffic forecasting	Dynamic adaptive spatial representation.
[20]	Shazeer *et al.* (2017)	Sparsely-gated mixture-of-experts layer with a learned gating network	General deep learning	The gate weights expert sub-networks for each input; not designed for graph-structured traffic data or for weighting spatial and temporal representations, but the basis of the gating principle used in GMDF-Net.
[10]	Chen *et al.* (2022)	Graph attention network with spatial-temporal clustering	Traffic-flow forecasting	Attention weights within one representation-learning architecture.
[11]	Qi *et al.* (2023)	Spatiotemporal graph network with fused attention mechanisms (STGNN-FAM)	Traffic-flow prediction	Attention-based weighting; used as the strongest baseline in this study.
[12]	Luo *et al.* (2023)	Dynamic graph convolution with attention fusion	Traffic-flow prediction	Dynamic graph-attention integration rather than component-level fusion.
[13]	Li *et al.* (2019)	Deep feature fusion model	Traffic-speed prediction	Feature-level fusion; contributions are not adapted to the observed traffic state.
[14]	Huang *et al.* (2019)	Deep dynamic fusion network	Traffic-accident forecasting	Dynamic fusion for a different task; no spatial, temporal and contextual component fusion for traffic state.

[15]	Wu *et al.* (2019)	Graph WaveNet: adaptive adjacency matrix with dilated causal convolution	Traffic forecasting	Learns the graph and temporal convolutions jointly; no separate input-dependent weighting of spatial and temporal representations.
[16]	Gu *et al.* (2021)	S4: structured state-space sequence model	Long-sequence modelling	Not designed for graph-structured traffic data; precursor of the selective state-space models used in the GMDF-Net temporal block.
[17]	Gu (2023)	Mamba: selective state-space sequence model	General sequence modelling	Not designed for graph-structured traffic data; provides the temporal mechanism that inspires the GMDF-Net temporal block.
[18]	Li *et al.* (2024)	STG-Mamba: selective state-space model for spatial-temporal graphs	Spatial-temporal graph forecasting	Integrates graph and state-space modelling within a block; spatial and temporal representations are not weighted by a separate gate.
[19]	Afandizadeh *et al.* (2024)	Review and comparison of deep learning and classical forecasting algorithms	Traffic forecasting	Comparative review; does not propose a fusion mechanism.
[21]	Du *et al.* (2020)	Dynamic transition CNN	Traffic-demand prediction	A single learned representation may vary in usefulness across traffic regimes.
[22]	Bai *et al.* (2020)	Adaptive graph convolutional recurrent network	Traffic forecasting	Adaptive graph learning; no input-dependent weighting of component outputs.
[23]	Lv *et al.* (2020)	Temporal multi-graph convolutional network	Traffic-flow prediction	Multiple spatial and temporal graphs; no dynamic fusion of separate neural components.
[24]	Zhang *et al.* (2024)	Survey of missing-data imputation	Traffic data quality	Addresses input quality, whereas GMDF-Net addresses representation fusion.
[25]	Yu et al. (2018)	STGCN: graph convolution with gated temporal convolution	Traffic forecasting	Fixed graph and fixed combination of spatial and temporal layers; baseline in this study and source of the PeMSD7(M) benchmark.
[26]	Kingma and Ba (2015)	Adam: adaptive moment estimation for stochastic optimisation	Model training	Not a forecasting model; the optimiser used to train GMDF-Net.
[27]	Hochreiter and Schmidhuber (1997)	Long short-term memory network	Sequence modelling	Models temporal dependence only, without spatial graph structure; baseline in this study.
[28]	Kipf and Welling (2017)	Graph convolutional network for semi-supervised learning	Graph-structured learning	Models spatial structure only, without temporal dynamics; baseline in this study and basis of the graph-convolution layers.

Abbreviations: GCN: graph convolutional network; CNN: convolutional neural network; GMDF-Net: Graph-Mamba Dynamic Fusion Network; DAGCRN: graph convolutional recurrent network with dynamic adjacency matrix; ADSTGCN: dynamic adaptive deeper spatio-temporal graph convolutional network; STGNN-FAM: spatiotemporal graph neural network with fusion of attention mechanisms; STGCN: spatio-temporal graph convolutional network; STG-Mamba: spatial-temporal graph Mamba; S4: structured state space sequence model; PeMSD7: Performance Measurement System district 7.

1.2.6 Adaptive Graph Learning and State-Space Sequence Models

Beyond fixed and time-varying graphs, adaptive graph learning infers spatial dependencies directly from data. Graph WaveNet [15] learns a self-adaptive adjacency matrix from trainable node embeddings and combines it with dilated causal convolutions, allowing the model to capture hidden dependencies that the predefined sensor graph does not encode. On the temporal side, structured state-space models [16] and, more recently, selective state-space models such as Mamba [17] provide linear-time sequence modelling in which the state-transition parameters depend on the input, so that the model can retain or discard information at each step. STG-Mamba [18] extended selective state-space modelling to spatial-temporal graph learning and evaluated it on benchmark forecasting datasets. These

approaches primarily integrate graph and sequence modelling inside a single block or architecture. GMDF-Net instead learns a spatial and a temporal representation in separate branches, using a Graph WaveNet-inspired adaptive-graph block and a Mamba-inspired state-space block, and weights them with an input-dependent gate. The blocks are custom implementations inspired by [15] and [17] and are not reproductions of the complete published architectures.

Table 1 shows that existing approaches mainly adapt the graph structure, learn attention weights, combine learned features within one model, or integrate graph and state-space modelling inside one block, whereas the relative contribution of separately learned spatial and temporal representations is not determined dynamically from the observed traffic state. The surveys [1, 3, 19], the structured state-space model [16], the mixture-of-experts layer [20], the optimiser [26] and the baseline models [25, 27, 28] are listed so that every cited work appears in the table.

1.2.7 Research Gap and Positioning of GMDF-Net

Existing graph-based, attention-based and fusion-based forecasters adapt the graph structure, learn attention weights within one architecture, or combine learned features in a fixed manner [4–13]. More recent adaptive-graph [15] and state-space [17, 18] models strengthen the spatial and temporal components but also integrate them in a fixed manner. The gap addressed here is therefore specific: whether the relative contribution of separately learned adaptive-graph and state-space representations can be set by an input-dependent gate, in the manner of mixture-of-experts gating [20], for traffic speed forecasting, and under which traffic conditions this helps. Among the studies reviewed in Table 1, none applies such a gate to separately learned adaptive-graph and state-space representations.

GMDF-Net treats an adaptive-graph spatial representation and a state-space temporal representation as complementary sources of predictive information and lets an input-dependent gate set their weights before the forecast is produced. The hypothesis, inherited from DNNF-ITS, is that the usefulness of the two representations varies with traffic conditions, so that adaptive weighting may outperform the individual branches and fixed fusion. This hypothesis has been examined for the predecessor only; for GMDF-Net it remains to be validated (Section 4).

Table 2 compares the main design choices of GMDF-Net with the two models that inspired it and with its predecessor.

Relation of the gate to attention, concatenation and mixture-of-experts fusion. Feature concatenation followed by a linear layer applies the same learned map to every input, so the relative influence of the two branches is fixed after training. Self-attention computes weights between positions of a sequence, whereas the gate of Eqs. (5) and (6) compares only the two representations of the same node and time step and returns two non-negative weights that sum to one. For $K = 2$ the softmax of Eq. (6) reduces to $\alpha_S = \sigma(g_S - g_T)$, a sigmoid of a learned score difference, so the gate is a dense soft routing between two experts, as in a mixture of experts [20] but without sparse selection or load balancing. GMDF-Net therefore does not introduce a new fusion operator; its contribution lies in applying input-dependent weighting to adaptive-graph and state-space branches for traffic-speed forecasting.

Table 2. Design Comparison of the Proposed Network with Graph WaveNet, Spatial-Temporal Graph Mamba and the Predecessor Fusion Model

Design aspect	Graph WaveNet [15]	STG-Mamba [18]	DNNF-ITS	GMDF-Net (this study)
Spatial modelling	Fixed supports plus learned adaptive adjacency	Graph module combined with a selective state-space model	2-layer GCN on the fixed distance graph	Fixed and learned adaptive adjacency (Graph WaveNet-inspired)
Temporal modelling	Dilated causal convolution	Selective state-space model	2-layer Bi-GRU	Mamba-inspired selective state-space module
Spatial–temporal integration	Interleaved in stacked blocks	Integrated within one architecture	Parallel branches, gated	Parallel branches, gated
Branch weighting	Not input-dependent	No separate gate reported	Softmax gate over three branches	Softmax gate over two branches
Contextual branch	No	No	Yes (MLP, 9 features)	No
Implementation	Published architecture	Published architecture	In-house (predecessor)	Custom blocks inspired by [15] and [17]

The gate adds $2d \cdot d_g + 3d_g$ parameters (W_g, b_g and the two vectors w_S, w_T), which is small compared with the spatial and temporal blocks. Its benefit relative to concatenation has so far been measured only for our earlier unpublished model (Section 3.7) and remains to be measured for GMDF-Net.

2. Materials and Methods

This section outlines the proposed Graph-Mamba Dynamic Fusion Network (GMDF-Net) framework for short- and medium-term traffic speed forecasting. Designed as an end-to-end predictive pipeline, the framework integrates data preprocessing, parallel spatial and temporal representation learning, dynamic feature fusion and target prediction. GMDF-Net is the successor of our earlier DNNF-ITS framework (Dynamic Neural Network Fusion for Intelligent Transportation Systems), which used GCN, Bi-GRU and contextual-MLP branches; the architectural differences are summarised in Table 4.

2.1 Dataset Description and Statistics

To evaluate the proposed framework under real-world conditions, the PeMSD7 (M) benchmark was used. PeMSD7(M) contains traffic speed (mph) recorded by 228 detector stations in District 7 of California through the Caltrans Performance Measurement System (PeMS) at 5-minute intervals over the weekdays of May and June 2012, as released with [25]. It is a traffic-speed dataset and should not be confused with the PeMS07 benchmark, which is commonly used for traffic-flow (volume) prediction with a different sensor set; all quantities reported in this study are speeds. Sensor spatial metadata were used to construct the fixed distance-based adjacency matrix $A \in R^{N \times N}$ for graph convolutional processing. Missing values in the continuous time-series logs were imputed using linear interpolation and historical mean strategies in accordance with standard preprocessing practices [24]. All input features were normalized using Z-score standardization prior to training. No external meteorological data were used in the experiments. Missing-data handling. The proportion of missing values in the PeMSD7 (M) speed records is 0.85% of all sensor–time-step entries. The imputation rule was applied separately to each sensor after the chronological 70:15:15 split, so that every partition was imputed on its own: (i) a gap of up to 3 consecutive 5-minute steps (15 minutes) was filled by linear interpolation between the nearest valid observations of the same sensor on either side of the gap inside the same partition; (ii) a longer gap, and any missing value at the start or end of a partition where valid observations exist on one side only, was filled with the historical mean of the same sensor for the same time of day and day of week, computed from the training partition only. Consequently, no imputed value in the training partition used any validation or test observation, the historical means used for the validation and test partitions were estimated from training data only, and the interpolation applied to the validation and test partitions used observations from within the respective partition only. Causality of the imputation. The interpolation in rule (i) is two-sided: for the validation and test partitions it uses the first valid observation after a gap, which is recorded after the forecast origin of any window that overlaps the gap. The preprocessing is therefore retrospective data cleaning and not a causal, real-time procedure. Because missing entries make up 0.85% of the data and only gaps of up to 15 minutes are interpolated, the effect on the aggregate metrics is expected to be small. It was not measured: the share of windows with interpolated inputs or targets was not recorded, and the evaluation was not repeated with causal imputation or with metrics restricted to observed targets. The results in Section 3 are accordingly reported as an offline evaluation. PeMSD7 (M) is a widely used benchmark from the Caltrans Performance Measurement System [25]. The traffic speed (mph) is measured at 5-minute intervals (12 intervals/hour) continuously. The statistical parameters of the data are summarized in Table 3.

Table 3. Summary Statistics of the Traffic Speed Dataset

Parameter	Value
Dataset	PeMSD7(M)
Location	District 7, California, USA
Number of sensors	228
Sampling interval	5 minutes
Prediction target	Traffic speed (mph)
Forecasting horizons	15, 30, and 60 minutes
Data split	Chronological train/validation/test split
Source	Caltrans PeMS, District 7; benchmark release of [25]
Period	May–June 2012, weekdays only
Missing entries	0.85% of sensor–time-step entries

Abbreviations: mph: miles per hour; PeMS: Performance Measurement System; PeMSD7(M): district 7 subset of PeMS with 228 sensors

The data set is split in chronological order into training, validation, and test sets in a ratio of 70:15:15. Normalization parameters are estimated only from the training partition and then applied unchanged to the validation and testing partitions to avoid information leakage; the input windows and prediction targets are then built separately for each partition in temporal order. The historical input sequences were created with a lookback window of 12 observations, equivalent to 60 minutes of history at 5-minute sampling. The forecasting horizons evaluated are 15, 30, and 60 minutes, equal to 3, 6 and 12 steps ahead. All baseline models were processed with the same sequence construction and preprocessing to allow a consistent comparison with GMDF-Net.

2.2 Spatial Graph Construction

The spatial adjacency matrix $A \in R^{N\times N}$ is constructed from the pairwise road-network distances d_{ij} between sensors with a thresholded Gaussian kernel (Eq. (1)). Closer sensors receive larger weights $w_{ij} = \exp\left(-d_{ij}^2/\sigma^2\right)$, and weak edges are discarded:

$$A_{ij} = \begin{cases} w_{ij}, & i \neq j \text{ and } w_{ij} \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Where d_{ij} is the road-network distance between sensors v_i and v_j, σ is a fixed distance bandwidth ($\sigma = 10$) and $\epsilon = 0.1$ is the sparsification threshold. Self-loops are added and the matrix is symmetrically normalised before graph convolution (Section 2.5.1).

Graph construction procedure. The adjacency matrix of the 228-sensor network was built as follows. (i) The pairwise road-network distance between each pair of the 228 PeMSD7 (M) sensors was obtained from the sensor distance metadata of the dataset, expressed in miles. (ii) For every pair of different sensors, the Gaussian weight of Eq. (1) was computed with a single global bandwidth σ = 10 (in the same unit as the distance); σ is a fixed constant and is not re-estimated for individual sensors. (iii) Weights below the sparsification threshold ε = 0.1 were set to zero, and weights greater than or equal to 0.1 were retained as real-valued edge weights, so the graph is weighted and not binarized. (iv) The diagonal was set to zero, self-loops were added, and the symmetric normalization of Section 2.5.1 was applied. Because the road-network distance is symmetric, the resulting matrix is symmetric; the graph is therefore undirected, and every retained edge is bidirectional with equal weight in both directions (the general graph formulation of Section 2.4 is directed). The graph is static: it is computed once from the distance matrix and is kept fixed during training and testing.

The fixed matrix A is static. In GMDF-Net it is complemented by a learned adaptive adjacency matrix A_{adp} (Section 2.5.1) that is trained jointly with the other parameters and is not constrained to follow road distance. The adaptive matrix is dense and is recomputed from the node embeddings at every parameter update, whereas A is computed once. The values $\sigma = 10$ and $\epsilon = 0.1$ are fixed; a sensitivity analysis of these parameters and a comparison of the fixed, adaptive and combined graphs are not reported in this study.

2.3 Contextual Feature Representations (Predecessor DNNF-ITS)

This subsection summarises the contextual branch of the predecessor DNNF-ITS, which GMDF-Net does not use. In DNNF-ITS the contextual representation combined a sine and cosine encoding of the hour of day (2 features) with a one-hot encoding of the day of week (7 features), giving $D_{\text{ctx}} = 9$ features derived from the timestamps alone, and was processed by a three-layer MLP with hidden sizes of 32, 64 and 64 and LeakyReLU activation. No external variables were used. The current GMDF-Net implementation has no separate contextual branch, so temporal and calendar information is not injected through a dedicated extractor.

2.4 Problem Formulation

Consider a spatial traffic network modelled as a graph $G = (V, E, A)$, where V is a set of N sensor nodes, E denotes the connectivity links and $A \in R^{N\times N}$ is the spatial adjacency matrix. At observation step t, the input $X_t \in R^{N\times F}$ contains F historical features per node. Given the sequence of the last L steps, $X_{t-L+1:t} \in R^{L\times N\times F}$, the task is to forecast the speed $Y_{t+h} \in R^{N\times 1}$ of every sensor at the horizons $h \in \{3,6,12\}$ steps.

2.5 System Architecture Overview

GMDF-Net processes the traffic tensor through two parallel branches. The spatial block (Section 2.5.1) learns node representations from the sensor graph and a self-learned adaptive graph; the temporal block (Section 2.5.2) models the recent history of each sensor with a selective state-space module. The two latent tensors $h_S, h_T \in R^{N\times d}$ are combined by the dynamic gating mechanism of Section 2.7 and mapped to the forecast by a prediction head. Throughout, d is the common hidden dimension of the two branches, d_e the node-embedding size, n_s the state size of the state-space module and d_g the hidden size of the gating layer; these symbols are listed in Table 6. The spatial and temporal blocks are custom implementations inspired by Graph WaveNet [15] and Mamba [17]; they are not the complete published architectures.

Figure 1 illustrates the overall framework of the proposed GMDF-Net approach.

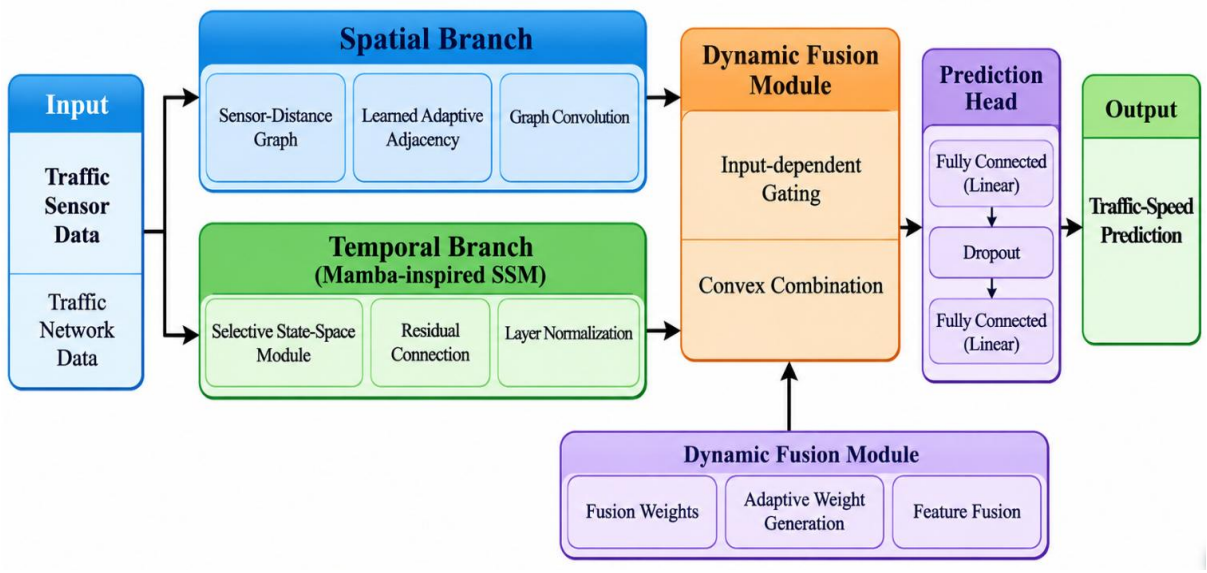


Figure 1. Overall GMDF-Net Framework.

The speed tensor and the sensor adjacency enter a spatial branch and a temporal branch in parallel, the two resulting representations are weighted by the dynamic gate, and a linear prediction head maps the fused representation to the forecasts at the three horizons; the equation numbers in the figure refer to Sections 2.5 to 2.7. The framework addresses non-stationary spatial-temporal dependencies through the two parallel sub-networks followed by an adaptive gating fusion layer. Compared with the predecessor DNNF-ITS, the GCN and Bi-GRU extractors are replaced by the adaptive-graph and state-space blocks and the contextual branch is not used.

2.5.1 Spatial Block (Adaptive-Graph Convolution)

To model topological graph structure, the spatial block takes the node-wise input $X_S \in R^{N\times(L\cdot F)}$ and combines two graph supports: the fixed distance-based support of Section 2.2, symmetrically normalised with added self-loops so that aggregation is not dominated by highly connected sensors, and a self-adaptive support learned from node embeddings, following the adaptive dependency matrix of Graph WaveNet [15]. The adaptive support (Eq. (2)) is computed from two node-embedding matrices:

$$A_{\text{adp}} = \text{SoftMax}\big(\text{ReLU}(E_1 E_2^{\top})\big) \qquad (2)$$

where $E_1, E_2 \in R^{N\times d_e}$ are learnable node embeddings. Because E_1 and E_2 are distinct, A_{adp} is in general asymmetric, i.e. it defines a directed graph. The spatial representation (Eq. (3)) is obtained by graph convolution over both supports:

$$h_S = \text{ReLU}\big(\textstyle\sum_m \hat{A}_m X_S W_m + b_S\big), \quad m \in \{\text{fix}, \text{adp}\} \qquad (3)$$

Here $\hat{A}_{\text{fix}}$ is the normalised fixed support, $\hat{A}_{\text{adp}} = A_{\text{adp}}$, W_m and b_S are learnable parameters and $h_S \in R^{N\times d}$. The adaptive support allows the model to represent dependencies that are not encoded by road distance, for example between sensors on parallel or connected corridors, while the fixed support preserves the physical prior; both are trained jointly with all other parameters. A dropout layer with rate $p = 0.2$ is applied within the block to prevent over-reliance on local spatial clusters.

2.5.2 Temporal Block (Mamba-Inspired State-Space Module)

The temporal block processes the sequence $X_T \in R^{N\times L\times F}$ of each node with a selective state-space module. A state-space model maps an input sequence x_τ to an output y_τ through a hidden state s_τ; after discretisation, the state is updated and read out as in Eq. (4),

$$\begin{aligned} s_\tau &= \bar{A}_\tau s_{\tau-1} + \bar{B}_\tau x_\tau \\ y_\tau &= C_\tau s_\tau \end{aligned} \qquad (4)$$

where, following the selection mechanism of Mamba [17], the hidden state $s_\tau \in R^{n_s}$ has state size n_s, and the step size and projection matrices depend on the input: $\Delta_\tau = \text{softplus}(W_\Delta x_\tau)$, $B_\tau = W_B x_\tau$, $C_\tau = W_C x_\tau$, $\bar{A}_\tau = \exp(\Delta_\tau A_S)$ and $\bar{B}_\tau \approx \Delta_\tau B_\tau$. Here A_s is the learnable state matrix and W_Δ, W_B and W_C are learnable projections. Because these parameters depend on the input, the module can retain or discard information at each step, which is useful when a speed series alternates between free-flow and congested behaviour, while its cost grows linearly with the sequence length L [17]. The output at the final step, y_L, is linearly projected to the temporal representation $h_T \in R^{N\times d}$.

The module is a custom Mamba-inspired implementation and not the hardware-aware selective-scan kernel of the official Mamba architecture; it is a

custom recurrence and does not use the official Mamba package.

2.6 Summary of Component Specifications

The architectural differences between the predecessor DNNF-ITS and GMDF-Net, together with the structural specification of the GMDF-Net sub-networks described in Section 2.5, are summarized in Table 4.

2.7 Dynamic Feature Fusion Mechanism

The latent outputs h_S and $h_T \in R^{N\times d}$ of the spatial and temporal branches are concatenated for every sensor and time step into $H_{\text{cat}} \in R^{2d}$, so that the gate can compare the two representations of the same sensor. The gating sub-network evaluates node- and time-dependent relevance scores g_k for the two branches $k \in \{S, T\}$ according to Eq. (5), where a tanh layer transforms the concatenated representation and a branch-specific vector scores it:

$$g_k = w_k^{\mathrm{T}} \tanh\left(W_g H_{\text{cat}} + b_g\right) \quad (5)$$

where $W_g \in R^{2d\times d_g}$, $b_g \in R^{d_g}$ and $w_k \in R^{d_g}$ are learnable parameters. Dynamic weights α_k are obtained by the softmax normalization of Eq. (6) over the two branches, so that they are positive and satisfy $\alpha_S + \alpha_T = 1$:

$$\alpha_k = \exp(g_k) / \textstyle\sum_j \exp\left(g_j\right) \quad (6)$$

The fused representation $H_F \in R^{N\times d}$ is the convex combination of the two representations given by Eq. (7):

$$H_F = \alpha_S h_S + \alpha_T h_T \quad (7)$$

A linear prediction head with learnable parameters then maps H_F to the forecast $\hat{Y}_{t+h}$, one value per sensor for each forecast horizon. Rather than enforcing rigid static weighting or naive unweighted summation, GMDF-Net applies a data-driven gating mechanism that assigns time-varying weights to the spatial and temporal representations based on the current input conditions. As shown in Figure 2, the fusion stage concatenates the two representations, scores them with a gating sub-network, normalises the scores with a softmax, forms a convex combination and projects it to the forecast.

2.7.1 Loss Function

The model parameters are optimized end-to-end using the Mean Absolute Error (MAE) loss of Eq. (8) on the normalised targets, which averages the absolute forecast error over all sensors and samples:

$$\mathcal{L}_{\text{MAE}} = (1/BN) \textstyle\sum_{b,n} \left|Y_{b,n} - \hat{Y}_{b,n}\right| \quad (8)$$

where B is the number of samples in a mini-batch, N is the number of sensors, and $Y_{b,n}$ and $\hat{Y}_{b,n}$ are the observed and forecast normalised speeds of sensor n in sample b.

Table 4. Architectural Comparison and Structural Specifications of the Predecessor and Proposed Models

Component	DNNF-ITS (predecessor)	GMDF-Net (proposed)	Input Shape	Layer Stack & Hidden Dimensions	Output Representation	Primary Functional Role
Spatial	2-layer GCN on the fixed distance graph	Adaptive-graph convolution (Graph WaveNet-inspired) over the fixed and adaptive supports	N × (L ·F)	Graph convolution over fixed and adaptive supports; width d, embedding size d_e	N × d	Road topology and learned latent spatial dependencies
Temporal	2-layer Bi-GRU	Mamba-inspired selective state-space module	N × L × F	Selective state-space recurrence with final-step readout	N × d	Sequential trends with input-dependent state dynamics
Contextual	3-layer MLP (9 features)	Not used	–	–	–	–
Fusion	Softmax gate over three branches (K = 3)	Softmax gate over two branches (K = 2)	N × 2d	tanh gating layer, softmax	N × d	Input-dependent weighting of branch outputs
Prediction head	Output projection	Prediction head	N × d	Linear projection to the forecast	N × 1 per horizon	Traffic-speed forecast

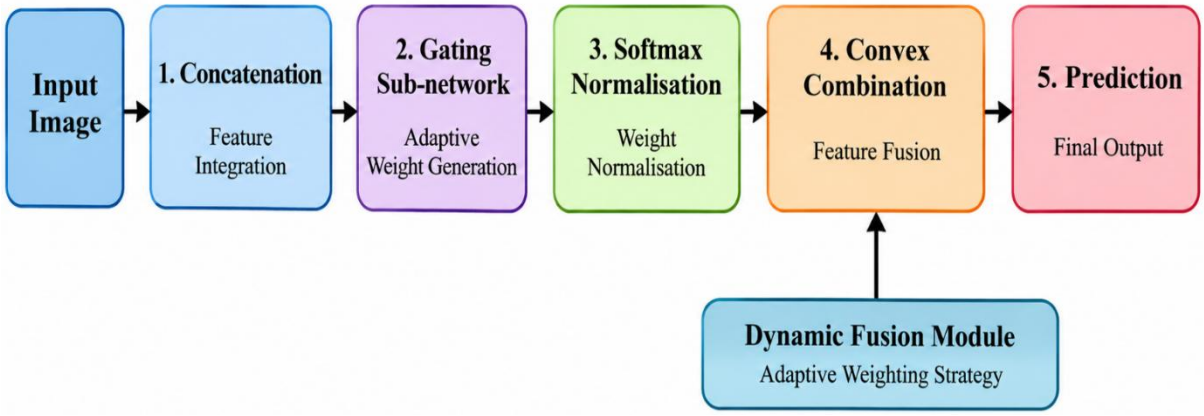


Figure 2. Dynamic Feature Fusion Architecture of GMDF-Net.

Table 5. Operational Training and Execution Hyperparameters

Parameter Name	Selected Value	Parameter Name	Selected Value
Optimization Algorithm	Adam	Early Stopping Criterion	Patience = 10 epochs
Initial Learning Rate	0.001 (1×10^{-3})	0.001	42
Mini-Batch Size	64	Data Split Ratio	Train (70%) / Val (15%) / Test (15%)
Maximum Epochs	50 (Figure 5)	Primary Training Loss	Mean Absolute Error (MAE)
Dropout Rate	0.2	Execution Platform	PyTorch / NVIDIA GPU (Google Colab)
Lookback Window (L)	12 steps (60 min)	Forecasting Horizons	3, 6, 12 steps (15, 30, 60 min)
Weight Decay / LR Scheduler	None	Gradient Clipping	5.0

Abbreviations: MAE: mean absolute error; LR: learning rate; GPU: graphics processing unit; min: minutes.

Table 6. Consolidated Implementation Details of the Proposed Model

Item	Specification
Input tensor	Size B × L × N × F with L = 12 and N = 228 (F input features per node)
Target	Traffic speed (mph), one value per sensor, horizons h = 3, 6 and 12 steps
Normalisation	Z-score, parameters from the training partition only
Fixed graph	Thresholded Gaussian kernel (Eq. (1)), σ = 10, ε = 0.1, symmetric normalisation with self-loops
Adaptive adjacency	Eq. (2); node-embedding size d_e
Spatial block	Hidden width d; dropout 0.2; Eq. (3)
Temporal block	State size n_s; Eq. (4); linear readout of the final step
Fusion gate	Hidden size d_g; Eqs. (5) to (7); softmax over the branch axis (K = 2) for every node and time step
Prediction head	Linear, one value per sensor for each horizon
Optimisation	Adam [26], learning rate 0.001, batch size 64, 50 epochs, gradient clipping 5.0, no weight decay or scheduler, MAE loss, seed 42
Checkpoint selection	Lowest validation MAE (Algorithm 1)
Software and hardware	PyTorch; NVIDIA GPU on Google Colab

Abbreviations: mph: miles per hour; MAE: mean absolute error; GPU: graphics processing unit.

2.8 Operational Configuration & Reproducibility Settings

To improve reproducibility, the principal model-training settings, dataset partitioning rules, and execution configuration are summarized in Table 5. GMDF-Net was trained for 50 epochs (Figure 5) with the Adam optimizer [26], an initial learning rate of 0.001, a mini-batch size of 64, MAE training loss, a dropout rate of 0.2, a fixed random seed of 42 and gradient clipping at 5.0; these settings were kept the same as in the earlier DNNF-ITS study. The dataset is partitioned chronologically into training, validation, and testing subsets using a 70:15:15 ratio. The structural specification of the sub-networks is given in Section 2.5 and Table 4. Table 6 consolidates the implementation details of GMDF-Net. The hidden dimensions of the blocks are denoted symbolically (d, d_e, n_s, d_g) as in Sections 2.5 to 2.7; parameter counts, training times and inference latency were not recorded (Section 4).

2.9 Algorithmic Execution Pipeline

The full end-to-end operational procedure of the proposed GMDF-Net framework—incorporating preprocessing, parallel spatial and temporal extraction, dynamic gating fusion and iterative backpropagation training—is formally structured in Algorithm 1.

Algorithm 1. End-to-End Training and Prediction Workflow of the Proposed Network

Inputs:

Raw dataset D, sensor adjacency matrix A

Lookback window $L = 12$, prediction horizon h, branches $K = 2$

Max epochs $E_{\max} = 50$, learning rate η = 0.001, mini-batch size B = 64, random seed $S = 42$

Outputs:

Predicted traffic states $\hat{Y}_{t+h}$, Evaluation metrics (MAE, RMSE, MAPE)

Step 1: Data Preprocessing

Load raw dataset D and order chronologically.

Clean data to obtain D_{clean} via quality control treatments.

Split D_{clean} chronologically into D_{train}, D_{val}, and D_{test} (70:15:15).

Compute normalization parameters ϕ using strictly D_{train}.

Normalize subsets: $D^N_{\text{train}}, D^N_{\text{val}}, D^N_{\text{test}} \leftarrow Normalize(D_{\text{train}}, D_{\text{val}}, D_{\text{test}}, \phi)$.

Form (X, Y) window pairs using lookback L and horizon h.

Step 2: Model Initialization

Initialize parameters $\Theta = \{\theta_S, \theta_T, \theta_{\text{fuse}}, \theta_{\text{pred}}\}$ with seed S.

Initialize tracking parameters: $L_{\text{best}} \leftarrow +\infty$ and patience counter $\leftarrow 0$.

Step 3: Training with Best-Validation Checkpointing

For epoch $= 1, \ldots, E_{\max}$ do:

For each mini-batch $(X_b, Y_b) \in D^N_{\text{train}}$ do:

$$A_{\text{adp}} \leftarrow SoftMax\left(ReLU(E_1 E_2^T)\right);\ h_S \leftarrow SpatialBlock(X_b, A, A_{\text{adp}}; \theta_S)$$

$$h_T \leftarrow TemporalSSM(X_b; \theta_T)$$

$$\{\alpha_S, \alpha_T\} \leftarrow DynamicGating(h_S, h_T; \theta_{\text{fuse}})$$

$$h_F \leftarrow \alpha_S h_S + \alpha_T h_T$$

$$\hat{Y}_b \leftarrow PredictionHead(h_F; \theta_{\text{pred}})$$

$$L_{\text{train}} \leftarrow MAE(\hat{Y}_b, Y_b)$$

Update parameters Θ via backpropagation.

Evaluate validation loss L_{val} on D^N_{val}.

If $L_{\text{val}} < L_{\text{best}}$ then:

$\Theta_{\text{opt}} \leftarrow \Theta$, $L_{\text{best}} \leftarrow L_{\text{val}}$, patience counter $\leftarrow 0$

Else: increase the patience counter by one and stop training when it reaches 10.

Step 4: Testing and Evaluation

Restore optimal parameters Θ_{opt}.

Generate normalized predictions on test set: $\hat{Y}_{\text{norm}} \leftarrow Predict(D^N_{\text{test}}; \Theta_{\text{opt}})$.

Rescale predictions: $\hat{Y}_{t+h} \leftarrow InverseNormalize(\hat{Y}_{\text{norm}}, \phi)$.

Compute metrics (MAE, RMSE, MAPE) comparing $\hat{Y}_{t+h}$ with unnormalized ground truth Y_{test}.

Return: Predicted traffic states $\hat{Y}_{t+h}$ and corresponding evaluation metrics.

2.10 Computational Complexity

The computational cost of the proposed GMDF-Net framework can be controlled by three major functional stages: representation extraction in the spatial and temporal blocks (Figure 1), coefficient evaluation in the dynamic fusion gating layer (Figure 2) and forward projection in the prediction head. The dense adaptive adjacency matrix grows quadratically with the number of sensors, the state-space recurrence grows linearly with the window length$L$, which is the property that motivates selective state-space models [17], and the gate adds only a weight matrix, a bias vector and two score vectors ($2d \cdot d_g + 3d_g$ parameters).For the $N = 228$ sensors of PeMSD7(M) the quadratic term is small, but it would dominate on much larger networks. Measured parameter counts, training time and inference latency were not recorded in the present experiments (Section 4), so no practical efficiency is claimed.

3. Results

This section empirically analyzes the proposed Graph-Mamba Dynamic Fusion Network (GMDF-Net). It compares GMDF-Net with conventional baselines and with the predecessor DNNF-ITS at three forecasting horizons using standard error metrics (Section 3.1), examines how its error grows with the horizon (Section 3.2), its training behaviour (Section 3.3) its forecasts at the sensor level (Section 3.4), the distribution of its errors (Section 3.5) and the variation of error across sensors (Section 3.6). Section 3.7 summarises the evidence reported for the predecessor DNNF-ITS, which motivates the dynamic fusion principle used in GMDF-Net.

3.1 Main Comparative Performance Analysis

The proposed framework is compared with Standard LSTM [27], Standard GCN [28], STGCN [25], STGNN-FAM [11] and the predecessor DNNF-ITS on three forecasting horizons: short term (h = 15 min), medium term (h = 30 min) and long term (h = 60 min), using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). The numerical results are reported in Table 7 and the MAE evolution over the forecasting horizon is shown in Figure 3.

Table 7. Performance Comparison Across Forecasting Horizons

Model Category	Model Name	Horizon (h)	MAE (mph)	RMSE (mph)	MAPE (%)
Baseline models (earlier study)	Standard LSTM [27]	15 min	3.42	5.81	7.82
		30 min	4.15	7.02	9.45
		60 min	5.28	8.94	12.10
	Standard GCN [28]	15 min	3.28	5.54	7.48
		30 min	3.98	6.78	9.12
		60 min	5.12	8.65	11.75
	STGCN [25]	15 min	2.85	4.82	6.51
		30 min	3.48	5.91	7.95
		60 min	4.45	7.58	10.15
	STGNN-FAM [11]	15 min	2.62	4.41	5.98
		30 min	3.21	5.48	7.32
		60 min	4.12	7.02	9.40
Predecessor (earlier study)	DNNF-ITS (earlier study)	15 min	2.21	3.75	5.05
		30 min	2.74	4.68	6.26
		60 min	3.51	6.01	8.02
Proposed (this study)	GMDF-Net (Ours)	15 min	2.29	4.33	5.45
		30 min	3.15	6.10	7.89
		60 min	4.27	8.11	11.48

Note: the GMDF-Net values are rounded to two decimals from the test-set evaluation in original speed units; all other entries are the values obtained in our earlier study and were not re-run under the GMDF-Net protocol. Abbreviations – MAE: mean absolute error; RMSE: root mean square error; MAPE: mean absolute percentage error; mph: miles per hour; min: minutes; LSTM: long short-term memory; GCN: graph convolutional network; STGCN: spatio-temporal graph convolutional network; STGNN-FAM: spatiotemporal graph neural network with fusion of attention mechanisms; DNNF-ITS: dynamic neural network fusion framework for intelligent transportation systems (predecessor model); GMDF-Net: Graph-Mamba Dynamic Fusion Network (proposed model).

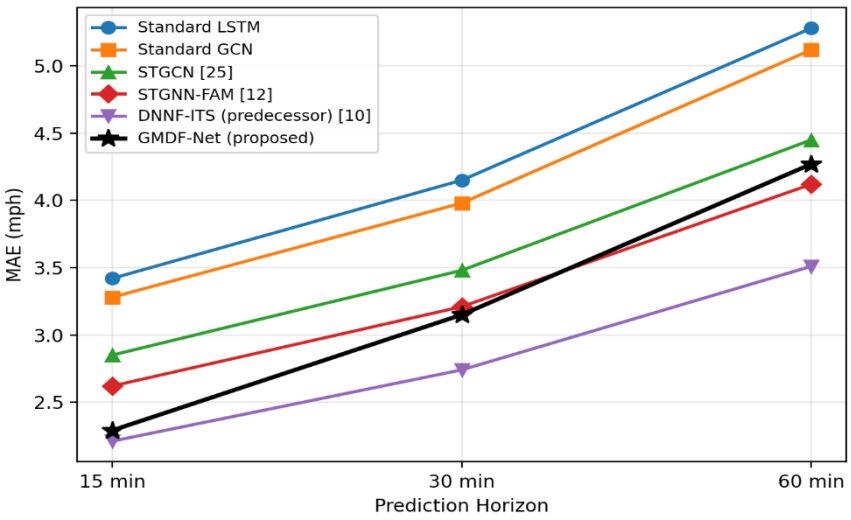


Figure 3. MAE Comparison across Forecasting Horizons.

Because the comparison combines values obtained in this study (GMDF-Net) with values obtained earlier (all other models), it is indicative and not a controlled benchmark.

The MAE, RMSE and MAPE are computed over all test observations in the original speed units (mph): the MAE averages the absolute error, the RMSE penalises large errors more strongly, and the MAPE expresses the absolute error relative to the observed speed. The treatment of zero or near-zero observed speeds in the MAPE was not documented, so the MAPE values should be interpreted with caution.

As shown in Table 7 and Figure 3, GMDF-Net achieves lower MAE than Standard LSTM, Standard GCN and STGCN at all three horizons. Relative to STGCN [25] the MAE reductions are 19.65%, 9.48% and 4.04% at 15, 30 and 60 minutes, respectively. Relative to the strongest external baseline, STGNN-FAM [11], GMDF-Net reduces the MAE by 12.60% at 15 minutes (2.29 vs 2.62 mph) and by 1.87% at 30 minutes (3.15 vs 3.21 mph), but its MAE at 60 minutes (4.27 mph) is 3.64% higher than that of STGNN-FAM (4.12 mph). The RMSE shows a less favourable pattern: GMDF-Net is marginally lower than STGNN-FAM at 15 minutes (4.33 vs 4.41 mph) but higher at 30 minutes (6.10 vs 5.48 mph) and at 60 minutes (8.11 vs 7.02 mph). The MAPE follows the MAE: at 15 minutes GMDF-Net (5.45%) is lower than STGNN-FAM (5.98%) and STGCN (6.51%), at 30 minutes it is lower than STGCN (7.89% vs 7.95%) but higher than STGNN-FAM (7.32%), and at 60 minutes it is higher than both (11.48% vs 9.40% and 10.15%). The ratio of RMSE to MAE for GMDF-Net (1.89, 1.94 and 1.90) is larger than that of STGNN-FAM (1.68, 1.71 and 1.70), which indicates that a larger share of its error is concentrated in a small number of large deviations; Sections 3.4 and 3.5 show that these correspond to abrupt speed transitions and to the upper tail of the error distribution. The relative MAE reduction is computed from the values in Table 7 as (MAE of reference − MAE of GMDF-Net) / MAE of reference × 100. Compared with the predecessor DNNF-ITS, the MAE of GMDF-Net is higher by 3.62%, 14.96% and 21.65% at 15, 30 and 60 minutes, so under the reported numbers the present configuration does not improve on the earlier three-branch model. As illustrated in Figure 3, the MAE of every model increase with the forecasting horizon, and the gap between GMDF-Net and the best models widens at the longer horizons.

3.2 Error Growth with Forecast Horizon

Figure 4 shows the test MAE and RMSE of GMDF-Net as a function of the prediction horizon, in the original dataset units (mph). The MAE increases from 2.29 mph at 15 minutes to 3.15 mph at 30 minutes and 4.27 mph at 60 minutes, i.e. by 37.6% and 35.6% for each successive step and by 86.5% overall. The RMSE grows from 4.33 mph to 6.10 mph and 8.11 mph (by 40.9% and 33.0%, and by 87.3% overall). Both metrics therefore grow approximately linearly with the horizon, and the RMSE remains close to 1.9 times the MAE throughout, with the absolute gap between the two curves widening from 2.04 to 2.95 and 3.84 mph. The widening gap indicates a heavier tail in the error distribution at longer horizons, which is examined in Section 3.5. This is consistent with a larger fraction of prediction windows containing a regime change, such as the onset or dissipation of congestion that cannot be inferred from a 60-minute history; squared error penalises these windows disproportionately. Section 3.4 examines this behaviour qualitatively.

3.3 Training and Validation Behaviors

Figure 5 shows the training and validation MAE, in normalised units, over the 50 training epochs of Adam optimisation [26].

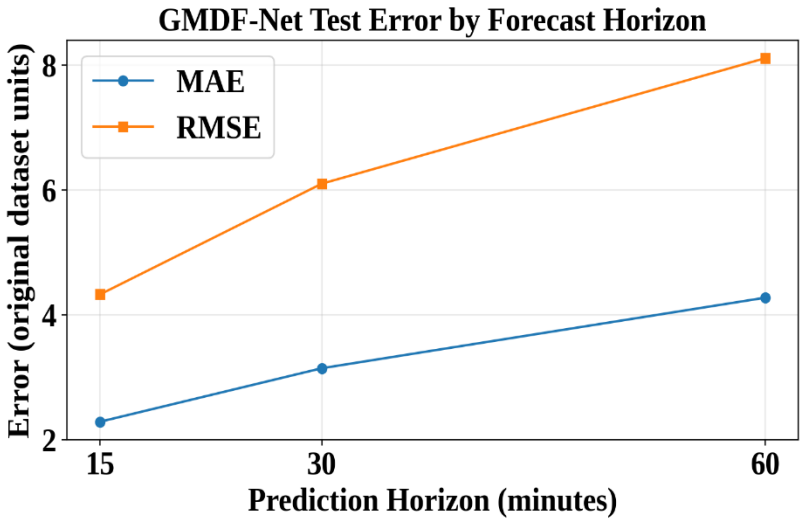


Figure 4. GMDF-Net Test Error (MAE and RMSE) by Forecast Horizon.

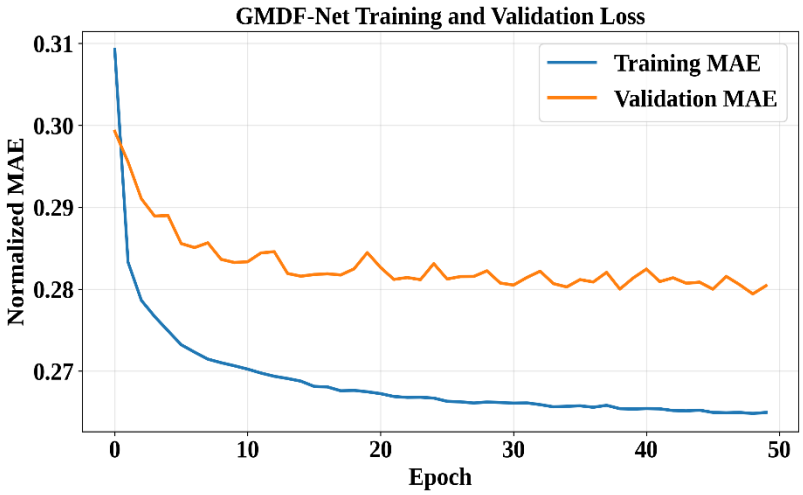


Figure 5. GMDF-Net Training and Validation Loss (Normalised MAE) over 50 Epochs.

Both curves fall rapidly in the first few epochs: the training MAE decreases from about 0.31 at epoch 0 to about 0.28 at epoch 1 and 0.27 by epoch 7, and then declines slowly to about 0.265 at epoch 49. The validation MAE decreases from about 0.30 to about 0.285 within the first ten epochs and then remains within 0.279–0.285 with fluctuations of the order of 0.003, reaching its lowest value of about 0.279 near epoch 48. Neither curve diverges and the validation error does not rise, so there is no sign of unstable optimisation or severe overfitting. However, a gap of about 0.015 normalised MAE separates training and validation error after epoch 20, and the validation curve is nearly flat while the training error continues to decrease. This indicates a moderate generalisation gap and suggests that further training would bring little benefit without stronger regularisation; weight decay and a learning-rate schedule were not used (Table 5). Because the validation curve oscillates around its plateau, the checkpoint with the lowest validation MAE (Algorithm 1) was used for testing. The training MAE at epoch 0 is slightly higher than the validation MAE (about 0.31 against 0.30), as expected when the training value is averaged over an epoch in which the weights are still improving. The values are in normalised units and cannot be compared directly with the mph values of Table 7.

3.4 Forecast Behaviour at the Sensor Level

Figure 6 compares the actual and predicted speed of Sensor 0 over 200 consecutive test windows at the 15-, 30- and 60-minute horizons. The sensor starts in a low-speed state (about 20–25 mph), rises to about 60–65 mph, falls back to a congested state of about 15–25 mph and then recovers to sustained free flow of about 65–75 mph after window 100. In the stationary periods the predictions follow the level of the series well: during sustained free flow the predicted speed is typically within about 5 mph of the actual speed, although it misses isolated spikes and dips of up to about 10 mph, and during the congested period it follows the low-speed

level while smoothing the fine-scale fluctuations, as expected from a model trained with an MAE objective.

The main weakness is visible at abrupt transitions. The predicted curve reproduces the shape of the actual transition but delayed. At the sharp recovery near window 100 the lag is about 3 windows at 15 minutes (Figure 6(a)), about 6 at 30 minutes (Figure 6(b)) and about 10–12 at 60 minutes (Figure 6(c)), and similar delays are visible for the earlier rise and for the drop after window 40. If consecutive test windows are separated by one 5-minute step, these delays are approximately equal to the forecast horizon, which suggests that around transitions the model behaves like a persistence forecaster, repeating the most recent state rather than anticipating the change. This is consistent with the larger RMSE-to-MAE ratio in Table 7 and the faster growth of error at 60 minutes, since a transition that occurs within the horizon contributes an error of up to 40–50 mph in a few windows. The first few windows of each plot also show over-prediction of roughly 15–17 mph (predicted about 39–41 mph against an actual 21–23 mph), which suggests an initial transient, probably a boundary effect of the test windowing, at the start of the test segment that was not examined further. Because only one sensor and 200 windows are shown, these observations are qualitative; moreover, Sensor 0 is the highest-error sensor at 60 minutes (Section 3.6), so the lags in Figure 6 should be read as a worst-case illustration rather than typical behaviour. Sections 3.5 and 3.6 extend the analysis to all test observations and all sensors. A last-value persistence baseline and a lag analysis, for example the cross-correlation lag between the predicted and actual series averaged over all sensors, are needed to quantify them.

3.5 Error Distribution and Tail Behaviour

The aggregate metrics of Table 7 summarise the error by its mean, so the distribution of the individual errors was examined on all test observations (Figure 7, Tables 8 and 9). The signed residuals in Figure 7(a) are centred close to zero at all three horizons, with a median that is nearly zero at 15 and 30 minutes and slightly negative (below 1 mph in magnitude) at 60 minutes, which indicates a mild tendency to over-predict at the longest horizon. The distributions are sharply peaked with very long tails on both sides, reaching about +48 and −52 mph at 15 minutes, +56 and −60 mph at 30 minutes and +61 and −62 mph at 60 minutes.

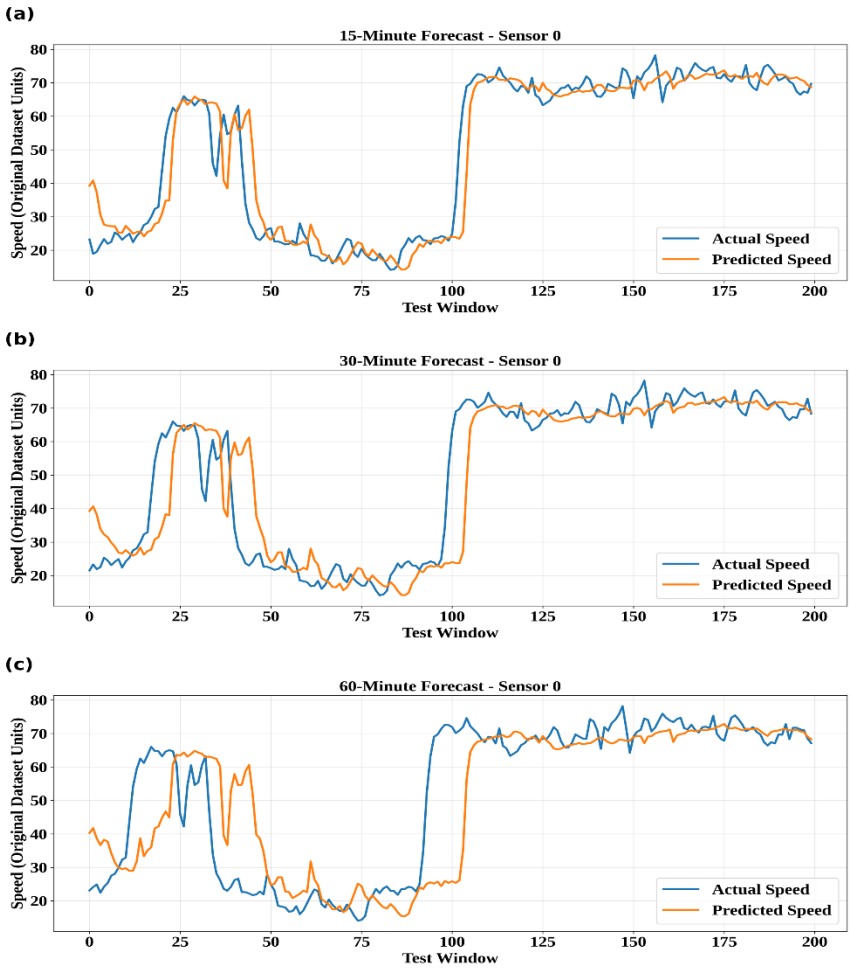


Figure 6. Actual versus predicted speed of Sensor 0 over 200 test windows: **(a)** 15-minute, **(b)** 30-minute and **(c)** 60-minute forecasts.

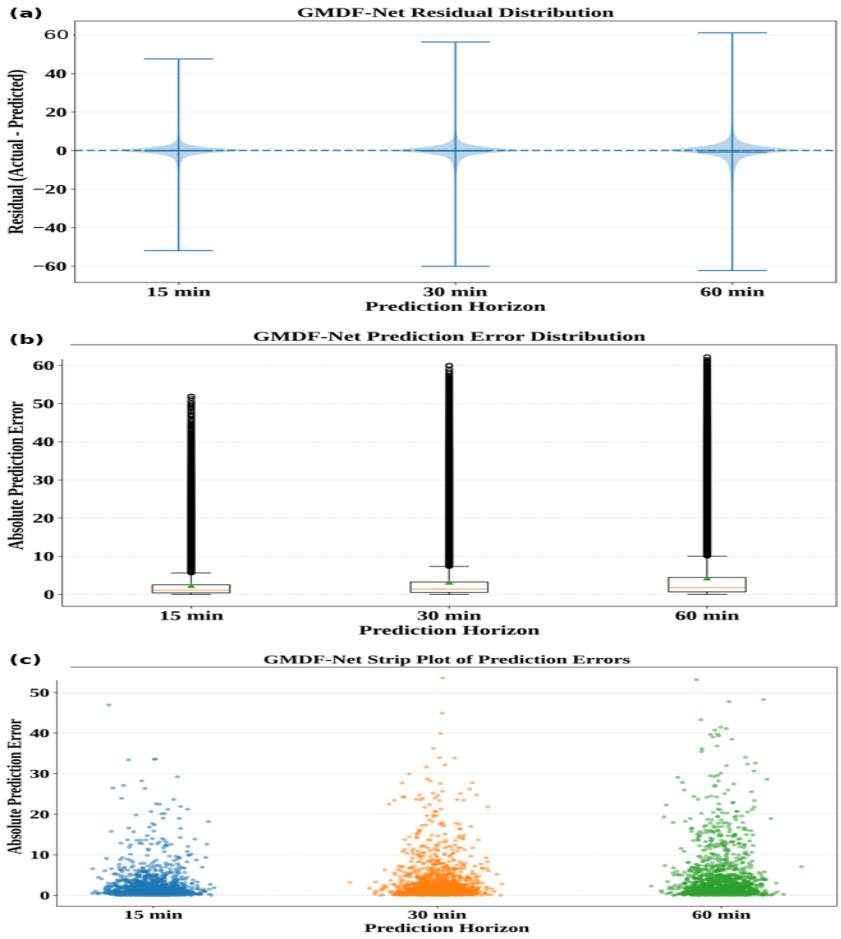


Figure 7. Distribution of GMDF-Net forecast errors on the test set at the 15-, 30- and 60-minute horizons: **(a)** signed residuals (actual − predicted) as violin plots, **(b)** absolute prediction errors as box plots with the mean marked by a triangle, and **(c)** strip plot of individual absolute errors (a subset of the test observations).

Because the extremes are of similar size in both directions, the large errors do not reflect a one-sided bias. They are consistent with the delayed response described in Section 3.4, in which the model over-predicts when speed drops and under-predicts when it recovers. The absolute errors in Figure 7(b) confirm that most errors are small and that the distribution is strongly right-skewed. The median absolute error is 1.12, 1.38 and 1.75 mph at 15, 30 and 60 minutes (Table 8), and the mean exceeds the median by a factor of 2.04, 2.28 and 2.45. The upper whiskers end at about 5.6, 7.3 and 10 mph, and the P90 values in Table 8 (5.35, 7.75 and 11.47 mph) lie close to these ends, so roughly one in ten observations is flagged as an outlier at each horizon. Because the outliers form a column too dense to be read in the box plot, the individual errors are shown in Figure 7(c): at every horizon most points lie below about 5 mph, whereas at 60 minutes many points lie between 10 and 40 mph and a few reaches about 50 mph, which is a much heavier upper tail than at 15 minutes.

Table 8 quantifies the tail. Between 15 and 60 minutes the median increases by a factor of 1.56, the P90 by 2.14 and the P99 by 1.80, so the upper tail grows faster than the typical error. At 60 minutes one forecast in ten is off by more than 11.47 mph and one in a hundred by more than 34.74 mph, which is comparable with the 40–50 mph difference between free-flow and congested speeds visible in Figure 6. The growth of MAE and RMSE with the horizon (Figure 4) is therefore largely associated with the upper tail of the error distribution and not by a uniform increase in all errors, which is also consistent with the RMSE-to-MAE ratio of about 1.9 discussed in Section 3.1.

Table 9 lists twenty individual test observations at the 15-minute horizon. Eighteen of them have an absolute error below 7 mph and the median of the listed errors is about 1.1 mph, close to the overall median of 1.12 mph in Table 8. Two observations (numbers 1 and 13) have errors of about 16 mph, which lie in the upper tail between the P95 and P99 values of Table 8. Observation 1 (actual 23.20 mph, predicted 39.24 mph) agrees with the start of the trace in Figure 6(a), which confirms that the over-prediction at the start of the plotted segment is a property of the model output and not a plotting artefact. The sample is too small to estimate error rates and serves only to show typical and extreme cases.

Table 8. Percentiles of the Absolute Prediction Error of the Proposed Model on the Test Set (Miles per Hour)

Horizon	Median (P50)	P90	P95	P99	Mean / median
15 min	1.12	5.35	8.84	19.33	2.04
30 min	1.38	7.75	13.27	27.61	2.28
60 min	1.75	11.47	19.31	34.74	2.45

Table 9. Twenty Test Observations of the Proposed Model at the 15-Minute Horizon (Speeds in Miles per Hour)

Obs.	Actual	Predicted	Abs. error	Obs.	Actual	Predicted	Abs. error
1	23.20	39.24	16.04	11	33.40	34.43	1.03
2	63.50	63.01	0.50	12	60.50	61.50	1.00
3	36.40	39.58	3.18	13	42.80	58.75	15.95
4	53.80	54.31	0.51	14	57.40	60.25	2.85
5	42.30	44.97	2.67	15	50.20	53.73	3.53
6	33.40	39.83	6.43	16	54.90	57.31	2.41
7	27.90	34.11	6.21	17	64.50	64.49	0.01
8	62.10	63.00	0.90	18	56.60	56.40	0.20
9	57.20	58.34	1.14	19	59.30	59.50	0.20
10	47.90	48.15	0.25	20	60.40	60.53	0.13

3.6 Sensor-Wise Error Analysis

Figure 8 shows the MAE of every sensor at the three horizons. The mean errors of Table 7 hide a wide spread across the 228 sensors: at 15 minutes the sensor MAE ranges from below 1 mph to about 5 mph, at 30 minutes up to about 7 mph and at 60 minutes up to about 9 mph, and at 15 minutes about four-fifths of the sensors have an MAE below 3 mph. The spread widens with the horizon, but the ordering of the sensors is almost unchanged: the rank correlation between the sensor-wise MAE at 15 and 60 minutes is about 0.9, sensors 0, 10, 171, 219 and 222 are among the ten highest-error sensors at all three horizons, and sensors 21, 54, 64, 110 and 111 are the five lowest-error sensors at all three horizons. Forecast difficulty is therefore largely a persistent property of the sensor. Sensor 0, which is used in Figure 6, has the highest MAE at 60 minutes (about 9 mph, roughly twice the network mean) and about 4.5 and 6.7 mph at 15 and 30 minutes, so Figure 6 illustrates a difficult case.

Figure 9 classifies the sensors into three groups of lower, medium and higher relative speed in the 15-minute evaluation (75, 78 and 75 sensors, respectively). The groups match the error pattern of Figure 8: the lower-speed group has about 1.6 times the MAE of the higher-speed group at 15 minutes (about 3.0 vs 1.9 mph) and about 2.0 times at 60 minutes (about 5.9 vs 3.0 mph), and 27 of the 30 highest-error sensors at 60 minutes, including all of the fifteen worst, belong to the lower-speed group. The runs of consecutive lower-speed sensors in Figure 9 (indices about 5–12 and 116–120) coincide with high-error bands in Figure 8. Sensors with lower relative speed, which are likely to spend more time in congested states, are therefore the hardest to forecast in this evaluation, which is consistent with the observation in Sections 3.4 and 3.5 that the large errors occur at missed transitions between free-flow and congested conditions.

3.7 Background Evidence from the Predecessor DNNF-ITS

Our earlier, unpublished DNNF-ITS model used a gated fusion of spatial, temporal and contextual representations. The results below are taken from that earlier work at the 30-minute horizon; they were not obtained in the present experiments and are not ablations of GMDF-Net. In its component ablation the full three-branch model with dynamic gating reached an MAE of 2.74 mph, compared with 3.02 mph without the contextual branch and 3.65 mph for the best single branch (temporal). Replacing dynamic gating by feature concatenation raised the MAE from 2.74 mph (RMSE 4.68 mph, MAPE 6.26%) to 2.96 mph (RMSE 5.02 mph, MAPE 6.75%). The average gate weights depended on the traffic regime: the temporal weight dominated in free flow (0.64 ± 0.07), the spatial and temporal weights were nearly equal in the transition state (0.43 and 0.42) and the spatial weight dominated under congestion (spatial weight 0.56 ± 0.08). This evidence motivates the dynamic gate retained in GMDF-Net, but it concerns a different architecture, and equivalent experiments for GMDF-Net are still required (Section 4).

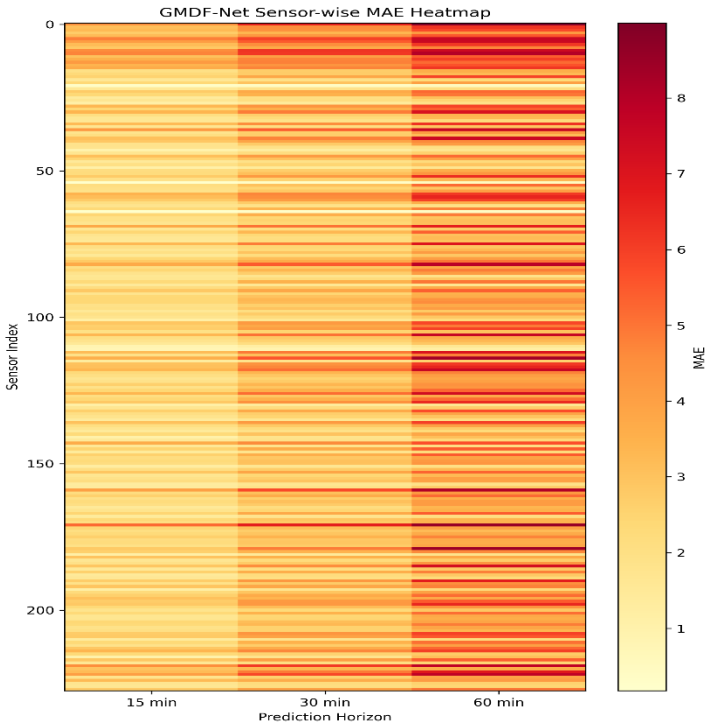


Figure 8. Sensor-wise MAE (mph) of GMDF-Net for the 228 sensors at the 15-, 30- and 60-minute horizons.

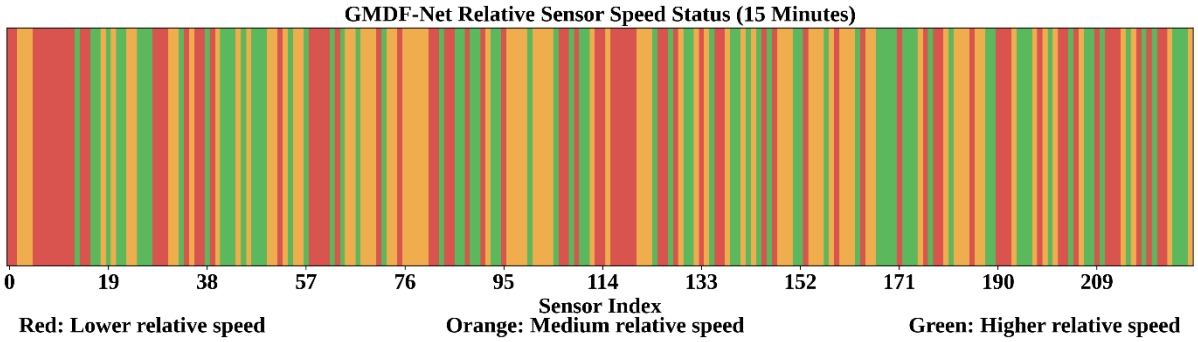


Figure 9. Relative speed status of the 228 sensors in the 15-minute evaluation (red: lower, orange: medium, green: higher relative speed).

4. Discussion

The results show that GMDF-Net improves on the conventional LSTM [27], GCN [28] and STGCN [25] baselines in MAE at all three horizons and on the strongest external baseline, STGNN-FAM [11], at the 15- and 30-minute horizons (by 12.60% and 1.87%). It does not improve on STGNN-FAM at 60 minutes, where its MAE is 3.64% higher, nor in RMSE at 30 and 60 minutes (Table 7). Its error grows approximately linearly with the horizon (Figure 4), and training is stable with a moderate generalisation gap (Figure 5).

The comparison with the predecessor is the most important finding for the direction of this work. On the reported numbers DNNF-ITS has a lower error than GMDF-Net at every horizon, and the difference grows with the horizon (3.62%, 14.96% and 21.65% in MAE). Several factors could contribute, and the present experiments do not separate them: (i) GMDF-Net omits the contextual branch which, in the ablation of the predecessor (Section 3.7), reduced the MAE from 3.02 to 2.74 mph at 30 minutes; (ii) the spatial and temporal blocks are custom, simplified versions of Graph WaveNet [15] and Mamba [17] components rather than the complete architectures; (iii) training used 50 epochs with settings that were not tuned for each horizon; and (iv) the baseline and DNNF-ITS figures come from an earlier set of runs and may not have been produced under exactly the same conditions. These factors can be tested by ablations of GMDF-Net (with and without the adaptive adjacency, with a GRU in place of the state-space module, with concatenation in place of gating, and with the contextual branch re-introduced) and by re-running all models under a single script.

The fusion analysis of the predecessor (Section 3.7) supports the principle that the usefulness of component representations varies with traffic state.

Dynamic gating reduced the MAE from 2.96 to 2.74 mph relative to feature concatenation, and the learned weights shifted from temporal-dominant in free flow (α_2 = 0.64 ± 0.07) to spatial-dominant under congestion (α_1 = 0.56 ± 0.08). Learned weighting of information has also helped in attention-based [11] and feature-fusion [13] forecasters, so a gain from gating is plausible. The weights describe the behaviour of the trained model and should not be read as causal evidence about traffic dynamics. Whether the two-branch gate of GMDF-Net shows the same regime-dependent behaviour has not yet been measured and is a priority for the next experiments.

The sensor-level forecasts (Figure 6) suggest that a major error source is the delayed response to abrupt speed transitions, with a lag of roughly one forecast horizon. A model that lags by the horizon at transitions resembles a persistence forecast at those points, and this limits the benefit of a more expressive spatial [15] or temporal [17, 18] block. Transition-aware training objectives (for example, loss weighting on large speed changes or a lag-penalising term), inputs that anticipate transitions (such as upstream-sensor speeds at longer lookbacks or event and calendar covariates) and evaluation with transition-specific metrics are therefore plausible directions to investigate before further increases in model size. The error distribution is strongly right-skewed, with a mean error two to two and a half times the median (Table 8), so the aggregate MAE is dominated by a minority of large errors, and the sensors with the largest errors almost all belong to the lowest relative-speed group (Figures 8 and 9). Sensor 0, shown in Figure 6, is the highest-error sensor at 60 minutes, so the lag in Figure 6 is a worst-case rather than a typical illustration.

Limitations affecting the validity of the present results. First, the GMDF-Net results come from a single run with a fixed seed; variability across seeds, confidence intervals and significance tests are not reported, and the 1.87% MAE difference from STGNN-FAM at 30 minutes is too small to be interpreted without repeated runs. Second, the baseline and predecessor results are carried over from an earlier study and may not share every preprocessing detail with the GMDF-Net run; no persistence or historical-average baseline is reported, and Graph WaveNet [15] and STG-Mamba [18], the models that inspired GMDF-Net, are not included as baselines. Third, the contributions of the dynamic gate, the adaptive adjacency and the state-space block have not been isolated by ablation for GMDF-Net, and the fusion evidence summarised in Section 3.7 belongs to the predecessor. Fourth, the evaluation is retrospective. Short gaps in the validation and test partitions were filled by linear interpolation between the valid observations on both sides of the gap (Section 2.1), so a small share of values, at most 0.85% of entries, may depend on observations recorded after the forecast origin, and imputed values may also appear among the targets used for the metrics. The reported errors therefore describe offline forecasting on cleaned data and should not be read as evidence of real-time operational performance. Fifth, the handling of zero or near-zero speeds in the MAPE was not documented, no sensitivity analysis of the graph-construction parameters is reported, and parameter counts, training time, inference latency, library versions and the GPU model are not reported, so no practical computational efficiency is claimed.

Limitations of scope. The evaluation uses a single benchmark, PeMSD7 (M) with 228 sensors, so generalisation to other networks, regions and measured quantities has not been tested; moreover, PeMSD7 (M) is commonly described as covering weekdays only [25]. The spatial and temporal blocks are custom implementations inspired by, not identical to, the published architectures. The time-series forecast plots show a single sensor, which is the highest-error sensor at 60 minutes, and the analysis of lag at abrupt transitions is qualitative. The study evaluates only the predictive-analytics component of an IoT-enabled ITS: IoT communication, sensing hardware and edge deployment are not implemented, and no external covariates such as weather or incident data are used. These points are addressed in the future work outlined in Section 5.

5. Conclusions

This study introduced GMDF-Net, a two-branch neural architecture for multi-horizon traffic speed forecasting in IoT-enabled intelligent transportation systems. The framework combines an adaptive-graph spatial branch, built from a fixed road-distance graph and a learned adjacency matrix, with a Mamba-inspired selective state-space temporal branch. An input-dependent gate integrates the two representations according to node- and time-specific information. On the PeMSD7 (M) benchmark, GMDF-Net achieved MAEs of 2.29, 3.15 and 4.27 mph, RMSEs of 4.33, 6.10 and 8.11 mph, and MAPEs of 5.45%, 7.89% and 11.48% for the 15-, 30- and 60-minute horizons, respectively.

The benchmark comparison indicates that GMDF-Net has lower MAE than the reported LSTM, GCN and STGCN results at all three horizons and then STGNN-FAM at 15 and 30 minutes. At 60 minutes, its MAE is 3.64% higher than STGNN-FAM, while the reported DNNF-ITS results remain lower across the evaluated horizons. These comparisons position GMDF-Net as a competitive approach while highlighting the importance of evaluating performance across multiple forecast ranges. The error analysis further identifies a strongly right-skewed distribution, greater errors at lower-speed sensors and delayed responses to abrupt speed transitions. Overall, the findings establish the predictive performance and benchmark characteristics of GMDF-Net across multiple forecasting horizons and

provide a foundation for further investigation of its fusion mechanism. Future work will quantify the individual contributions of the spatial branch, temporal branch and adaptive gate through component-wise ablations, repeated runs with multiple random seeds and protocol-matched comparisons. Evaluation across additional datasets, traffic regimes and operational settings will further assess the model's robustness and practical applicability.

References

[1] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, Y. Wang, Dynamic neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence, IEEE, 44(11), (2022) 7436–7456. https://doi.org/10.1109/TPAMI.2021.3117837

[2] S. Guo, Y. Lin, H. Wan, X. Li, G. Cong, Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Transactions on Knowledge and Data Engineering, IEEE, 34(11), (2022) 5415-5428. https://doi.org/10.1109/TKDE.2021.3056502

[3] K.H.N. Bui, J. Cho, H. Yi, Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. Applied Intelligence, 52(3), (2022) 2763-2774. https://doi.org/10.1007/s10489-021-02587-w

[4] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, Y. Li, Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. ACM Transactions on Knowledge Discovery from Data, 17(1), (2023) 1-21. https://doi.org/10.1145/3532611

[5] Z. Shi, Y. Zhang, J. Wang, J. Qin, X. Liu, H. Yin, H. Huang, DAGCRN: Graph convolutional recurrent network for traffic forecasting with dynamic adjacency matrix. Expert Systems with Applications, 227, (2023) 120259. https://doi.org/10.1016/j.eswa.2023.120259

[6] Y. Xu, L. Han, T. Zhu, L. Sun, B. Du, W. Lv, Generic dynamic graph convolutional network for traffic flow forecasting. Information Fusion, 100, (2023) 101946. https://doi.org/10.1016/j.inffus.2023.101946

[7] Z. Li, J. Zhou, Z. Lin, and T. Zhou, Dynamic spatial aware graph transformer for spatiotemporal traffic flow forecasting. Knowledge-based systems, 297, (2024) 111946. https://doi.org/10.1016/j.knosys.2024.111946

[8] Z. Xia, Y. Zhang, J. Yang, L. Xie, Dynamic spatial-temporal graph convolutional recurrent networks for traffic flow forecasting. Expert Systems with Applications, 240, (2024) 122381. https://doi.org/10.1016/j.eswa.2023.122381

[9] Z. Cui, J. Zhang, G. Noh, H.J. Park, ADSTGCN: A dynamic adaptive deeper spatio-temporal graph convolutional network for multi-step traffic forecasting. *Sensors*, 23(15), (2023) 6950. https://doi.org/10.3390/s23156950

[10] Y. Chen, T. Shu, X. Zhou, X. Zheng, A. Kawai, K. Fueda, Z. Yan, W. Liang, K.I.K. Wang, Graph attention network with spatial-temporal clustering for traffic flow forecasting in intelligent transportation system. IEEE Transactions on Intelligent Transportation Systems, 24(8), (2022) 8727–8737. https://doi.org/10.1109/TITS.2022.3208952

[11] X. Qi, W. Hu, B. Li, K. Han, STGNN-FAM: A traffic flow prediction model for spatiotemporal graph networks based on fusion of attention mechanisms. Journal of Advanced Transportation, 2023(1), (2023) 8880530. https://doi.org/10.1155/2023/8880530

[12] X. Luo, C. Zhu, D. Zhang, Q. Li, Dynamic graph convolutional network with attention fusion for traffic flow prediction. In: ECAI 2023: 26th European Conference on Artificial Intelligence, September 30–October 4, 2023, Kraków, Poland – Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023). SAGE Publications, (2023) 1633-1640. https://doi.org/10.3233/FAIA230446

[13] L. Li, X. Qu, J. Zhang, Y. Wang, B. Ran, Traffic speed prediction for intelligent transportation system based on a deep feature fusion model. Journal of Intelligent Transportation Systems, 23(6), (2019) 605-616. https://doi.org/10.1080/15472450.2019.1583965

[14] C. Huang, C. Zhang, P. Dai, L. Bo, Deep dynamic fusion network for traffic accident forecasting. In Proceedings of the 28th ACM international conference on information and knowledge management, (2019) 2673-2681. https://doi.org/10.1145/3357384.3357829

[15] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph WaveNet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121. https://doi.org/10.48550/arXiv.1906.00121

[16] A. Gu, K. Goel, C. Re, (2021). Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396. https://doi.org/10.48550/arXiv.2111.00396

[17] A. Gu, (2023). Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint, arXiv:2312.00752. https://arxiv.org/pdf/2312.00752

[18] L. Li, H. Wang, W. Zhang, A. Coster, (2024). STG-Mamba: Spatial-temporal graph learning via selective state space model. arXiv preprint arXiv:2403.12418. https://doi.org/10.48550/arXiv.2403.12418

[19] S. Afandizadeh, S. Abdolahi, H. Mirzahossein, Deep learning algorithms for traffic forecasting: A comprehensive review and comparison with classical ones. Journal of Advanced

Transportation, 2024(1), (2024) 9981657. https://doi.org/10.1155/2024/9981657

[20] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, J. Dean, (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538. https://doi.org/10.48550/arXiv.1701.06538

[21] Du, X. Hu, L. Sun, J. Liu, Y. Qiao, W. Lv, Traffic demand prediction based on dynamic transition convolutional neural network. IEEE Transactions on Intelligent Transportation Systems, 22(2), (2020) 1237-1247. https://doi.org/10.1109/TITS.2020.2966498

[22] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems, 33, (2020) 17804-17815.

[23] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, S. Ji, Temporal multi-graph convolutional network for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 22(6), (2020) 3337-3348.

[24] Y. Zhang, X. Kong, W. Zhou, J. Liu, Y. Fu, G. Shen, A comprehensive survey on traffic missing data imputation. IEEE Transactions on Intelligent Transportation Systems, 25(12), (2024) 19252-19275. https://doi.org/10.1109/TITS.2024.3478816

[25] B. Yu, H. Yin, Z. Zhu, (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875. https://doi.org/10.48550/arXiv.1709.04875

[26] D.P. Kingma, J. Ba, (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

[27] S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural computation, 9(8), (1997) 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[28] T.N. Kipf, M. Welling, (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. https://doi.org/10.48550/arXiv.1609.02907

Authors Contribution Statement

D. Sivabalaselvamani: Conceptualization, Methodology, Formal analysis, Writing-original draft. D. Thiyagarajan: Methodology, Investigation. M. Menagadevi: Formal analysis. S. Yasotha: Validation, Writing – review & editing. K. Nanthini: Validation, Writing – review & editing. Sankar Ganesh S: Writing – review & editing. All authors have read and approved the final version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License

© The Author(s) 2026. The text of this article is open access and licensed under a Creative Commons Attribution 4.0 International License.