

DEEP FUZZY NEURAL INTELLIGENCE FOR TRAFFIC PATTERN MINING AND ADAPTIVE PREDICTION IN INTELLIGENT TRANSPORTATION SYSTEMS

Mohammad Abdur Rasheed

Department of Civil Engineering, College of Engineering, Shaqra University, Saudi Arabia

Abstract

In this paper, a new deep fuzzy network structure called Fuzzy Temporal Graph Neural Intelligence Network (FT-GNIN) is designed for traffic patterns mining and intelligent transportation prediction. The presented method involves adaptive fuzzy membership functions, multiscale temporal features extraction, learning the spatial dependencies with graphs and attention guided neural fusion. First of all, observed traffic data is normalized and converted into time-series. Afterwards, using fuzzy adaptive inference, uncertain traffic data is converted into fuzzy membership functions representation. In parallel, spatial dependencies are captured by the graph neural module, while temporal dependencies are identified by the temporal learning module. Attention guided fusion is applied for merging the three types of information and generating the prediction of traffic flow, congestion and travel times. At the evaluation step 30, FT-GNIN has the following performance metrics: MAE=2.58 mph, RMSE=4.42 mph, MAPE=7.35%, $R^2=0.94$, and congestion classification F1-score=0.97. Compared with Graph WaveNet, our approach achieves relative improvements of 8.51% in MAE, 6.95% in RMSE and 8.13% in MAPE, as well as an absolute improvement of 0.03 in R^2 and 0.01 in F1-score.

Keywords:

Deep Fuzzy Neural Network, Traffic Pattern Mining, Intelligent Transportation Systems, Spatiotemporal Prediction, Adaptive Congestion Forecasting

1. INTRODUCTION

The concept of intelligent transportation system (ITS) has become inevitable for the management of urban mobility, alleviation of congestion, effective road usage, and evidence-based decision-making in transportation. Urbanization, increase in vehicle population, and complexity of road networks have led to the necessity of estimating traffic states and predictive traffic management. Modern infrastructure of transportation continuously collects data from various devices such as loop detectors, GPS equipped vehicles, roadside sensors, surveillance cameras, and transport network. Observations from these sources include the details of traffic flow, vehicle speed, road occupancy, travel time, and congestion level. Based on these details, intelligent prediction methods can be used to determine traffic trends and predict the changes in road networks before severe congestion happens. According to previous researches, deep learning and graph-based learning can be efficient methods for representation of complicated traffic interactions and prediction of transportation systems [1–3].

Traffic pattern mining refers to the identification of meaningful patterns within historical and real-time data about traffic such as periodicity, rush hour features, spread of traffic congestion, and other abnormalities in traffic behavior. Traditional statistical methods, like the autoregressive models, can make effective predictions only if the patterns are stable. Their capacity to capture highly nonlinear patterns and complex

interactions between distributed road segments may not be sufficient. The machine learning techniques came later to offer nonlinear feature learning, and deep neural networks offered the capability to automatically identify temporal representations from massive traffic data. Graph neural networks further enhanced this capability by modeling road segments as inter-connected nodes and traffic relations as graph edges. Therefore, combining fuzzy inference with deep spatiotemporal learning seems to be a promising approach towards traffic prediction.

Even though there have been considerable improvements in traffic forecasting models, there are still several challenges that are impacting the reliability of predictions. Firstly, traffic data often includes noise due to the malfunctions of sensors, missing data, irregular time sampling, changes in weather, and unusual driving behavior. Traditional neural networks tend to consider such observations as exact numerical values that might limit their capability of modeling the ambiguity in traffic states. In contrast to that, fuzzy logic is another technique used for traffic modeling since it provides an alternative description of the state in traffic in terms of such linguistic categories as low, medium, heavy, and congested. However, such fuzzy membership functions remain static and thus fail to reflect possible changes in the traffic distribution. Secondly, traffic conditions are rather complicated and include temporal and spatial dependencies. Traffic congestion in one particular intersection tends to spread to other road segments, and regular daily and weekly patterns intersect with accidents, roadworks, weather, and special events. Thus, a predictive model should learn both local temporal changes and interactions in the network.

Incorporating fuzzy inference, graph learning, and temporal pattern extraction raises new issues concerning model complexity, parameter tuning, uncertainty propagation, and stable learning in various traffic scenarios. Current traffic prediction methods mostly emphasize temporal modeling or spatial relationship learning but lack an approach of incorporating adaptive fuzzy uncertainty modeling into their architectures. Therefore, it becomes a major issue to simultaneously incorporate uncertain traffic states, network relationships, and temporality at multiple scales into one model. Insufficient interpretability may further limit the applicability of prediction results in real-world transport operations. The key issue this study aims to solve is how to devise a computationally efficient and uncertainty-aware deep learning architecture that can extract useful traffic patterns and make predictions on future traffic conditions in interconnected road networks [6].

The goal of this research is to design a deep fuzzy neural network framework for mining traffic patterns and intelligent transportation prediction.

The suggested Fuzzy Temporal Graph Neural Intelligence Network (FT-GNIN) offers a unified architecture where adaptive fuzzy reasoning, multi-scale temporal learning, graph-embedded

spatial representation, and feature fusion guided by attention mechanisms are combined. Contrary to a standard neural architecture that directly maps numerical observations to predictions, FT-GNIN explicitly captures uncertain traffic conditions via learnable fuzzy membership functions and then incorporates them into the learned spatiotemporal features. The key methodological novelty is the combination of fuzzy representations taking into account uncertainties and dynamically weighted road-network relations within the common predictive learning model. It is aimed at enhancing the robustness of the model for the case of uncertain traffic observations and evolving congestion patterns. The research presents an integrated fuzzy and graph-temporal learning architecture that considers uncertain traffic data, temporal dependencies and spatial interrelations of adjacent road sections. It uses trainable membership functions and graph embedding along with multi-scale temporal learning in order to mine traffic patterns.

2. RELATED WORKS

Advances in the field of intelligent transportation systems prediction modeling have evolved from simple statistical time-series prediction into advanced deep neural network architectures that are able to learn non-linear temporal and spatial relations and more complicated traffic patterns. Although the existing methodologies present a solid basis for designing the FT-GNIN approach, uncertainty modeling and fuzzy reasoning combined with adaptive spatiotemporal learning are still open questions.

[7] analyzed the deep learning approaches for traffic prediction modeling and highlighted the significance of learning complicated spatiotemporal relations in transport systems. The survey addressed neural architectures, traffic prediction applications, data sets, and evaluation procedures used in the research area. Deep learning architectures present an advantage over traditional statistical techniques since they allow automatic learning of non-linear relations based on historical observations. Nevertheless, the effectiveness of predictions depends on modeling of spatial relations, sufficient training data set, and generalization capabilities of the model.

[8] provided an extensive review of the application of graph neural networks for traffic forecasting. The survey examined graph convolution and graph attention neural network techniques applied to road traffic, public transport, and demand forecasting. Graphs are especially suitable to be used for transportation systems due to the fact that road segments and intersections form interdependent networks in which traffic is spatially dependent. Graph neural networks spread information among neighboring nodes, which helps the forecasting algorithms discover dependencies that traditional independent time-series forecasting models are unable to detect. Yet the effectiveness of graph-based models depends on their ability to construct graphs correctly and capture traffic-related dependencies.

Spatiotemporal Graph Convolutional Network (STGCN) [9] merges graph and temporal convolutions to model traffic dynamics. While graph convolution identifies the spatial dependencies between adjacent road segments, temporal convolution learns how traffic changes over time in observation sequences. The architecture offers an effective alternative to purely recurrent neural networks-based approaches. The parallel

nature of the network also allows for efficient implementation. Yet traditional approaches to STGCN implementation rely on predefined spatial relations and numerical representation of features. Adaptive fuzzy memberships could add another dimension to traffic measurements representation.

Proposed in [10] was DCRNN, where traffic flow propagation is modeled based on graph diffusion and recurrent temporal learning. Diffusion process in DCRNN reflects the directional relationships between the connected road segments, whereas recurrent process captures the temporal dependency between the data points. The method can be very useful for modeling the propagation of congestion on road networks where the upstream/downstream effect does not necessarily have to be symmetric. Unfortunately, recurrent process involves some computational complexity, and pre-defined diffusion relationships cannot reflect the dynamic interactions between traffic flows.

Graph WaveNet [11] is the combination of graph convolution, temporal convolution and the adaptive dependency matrix that allows for discovering relationships that cannot be captured by the initial road network graph. This is important since proximity of locations does not imply their traffic similarity. Two locations that are far away from each other can be correlated in traffic patterns due to commuting or road network topology. Therefore, adaptive graph learning provides a good starting point for traffic prediction improvement. However, learned dependencies are not interpretable in terms of uncertainty descriptions. Fuzzy representation can help adaptive graph learning since it represents membership degrees of traffic observations in linguistic states.

[12] Attention mechanisms were introduced into traffic prediction architectures to give different weights to the observations, time periods and locations. Attention-based spatial-temporal graph convolutional network (ASTGCN) created by Guo et al. applies spatial and temporal attention along with the graph convolution in order to model dynamic traffic dependencies. Attention mechanisms help to detect influential locations and traffic observations.

[13] The graph multi-attention network (GMAN), proposed by Zheng and colleagues, employs multiple attention mechanisms to represent spatial and temporal relationships in network-wide traffic forecasting. Its encoder-decoder design supports predictions across multiple future time steps and can model interactions between historical and future traffic representations. This approach demonstrates the value of attention for long-range traffic prediction. Nevertheless, complex attention architectures may require substantial computational resources, and their predictions can remain sensitive to data quality and distribution changes. Integrating fuzzy membership representations could provide complementary information about uncertain traffic states and improve the interpretability of intermediate features.

[14] surveyed the graph neural network frameworks for traffic prediction that include graph convolutional networks, graph attention networks, graph autoencoders, and graph generative models. The proposed classification is informative as it reflects the diversity of approaches for transportation research based on graphs. Relevant works point out the necessity of multi-source data fusion, model generalization and evaluation in various transportation scenarios. The results show that traffic prediction is more complicated than simply increasing the model depth and

requires taking into account the relevant connections, variable nature of traffic, and computational complexity.

[15] In recent literature reviews, the hybrid traffic prediction architectures that are built based on the combination of graph neural network techniques and attention mechanisms, dynamic graph learning, feature extraction were studied. The modern approaches are focused on the accounting for spatial heterogeneity, variable traffic dependencies, and multi-source information. However, the combination of graph learning and deep neural networks does not ensure explicitly modeling of ambiguous traffic states. Likewise, the fuzzy neural systems, which focus on representing uncertainties, do not necessarily incorporate spatial and temporal dependencies throughout the network.

3. PROPOSED METHODOLOGY

Proposed FT-GNIN will be employed to conduct traffic pattern analysis with the capability of uncertainty handling, as well as intelligent transportation system prediction via adaptive fuzzy inference, multiscale temporal features extraction, graph-based dependency among adjacent roads learning and attention-driven feature fusion. Specifically, traffic data collection from road sensors, GPS equipped cars and transportation system surveillance system will be conducted, as well as traffic data cleaning, normalization, dealing with missing values and temporal window building. Adaptive fuzzy membership functions will convert numerical traffic data into representation of uncertain traffic situation. Afterwards, temporal learning component will extract short-term fluctuation and long-term traffic trend. Meanwhile, graph neural component will learn dependency relationship among adjacent roads. Finally, attention-driven feature fusion will integrate above mentioned components for traffic prediction, traffic congestion classification and traffic travel time prediction.

4. PROBLEM FORMULATION AND TRAFFIC DATA REPRESENTATION

The first stage of FT-GNIN represents the transportation network as a graph whose nodes correspond to road segments, intersections, or monitored traffic locations. The graph edges describe physical connectivity, directional traffic relationships, or empirically established correlations between traffic observations. Let the transportation network be represented by $G = (V, E, A)$, where V denotes the set of N road nodes, E represents the edge set, and $A \in \mathbb{R}^{N \times N}$ denotes the weighted adjacency matrix. Each node is associated with measurements such as traffic flow, vehicle speed, occupancy, and travel duration. These variables provide complementary information for identifying congestion propagation and recurrent traffic behavior. The traffic observations are organized into fixed-length temporal windows so that the model receives both recent traffic conditions and historical information necessary for prediction.

Let $X_t \in \mathbb{R}^{N \times F}$ denote the feature matrix at time t , where F is the number of measured or derived traffic features. The historical input tensor is constructed from T consecutive observations, while the target represents traffic conditions over

the next H prediction intervals. This formulation supports multi-step forecasting and enables the same architecture to be evaluated at different prediction horizons. The input variables can include normalized traffic volume, average speed, road occupancy, time-of-day encoding, day-of-week indicators, and available contextual measurements. The target variables may represent continuous traffic flow and travel time or discrete congestion categories. The graph topology and temporal input windows remain consistent across training, validation, and testing to prevent information leakage.

The mathematical representation of the input traffic sequence is defined as $X_{t-T+1:t} = [X_{t-T+1}, X_{t-T+2}, \dots, X_t] \in \mathbb{R}^{T \times N \times F}$, where T represents the historical observation length, N denotes the number of monitored road nodes, and F identifies the feature dimension. The temporal tensor preserves the ordering of observations, allowing the learning modules to identify recurring traffic patterns, transient fluctuations, and longer-term variations. The corresponding prediction target is represented as $Y_{t+1:t+H} = [Y_{t+1}, Y_{t+2}, \dots, Y_{t+H}] \in \mathbb{R}^{H \times N \times Q}$, where H denotes the prediction horizon and Q represents the number of output variables. For example, Q can include traffic flow, vehicle speed, and travel time. When congestion classification is required, a separate categorical output head produces class probabilities for each road node and prediction interval. This multi-output formulation enables FT-GNIN to learn shared traffic representations while retaining task-specific prediction layers.

4.1 TRAFFIC DATA PREPROCESSING AND ADAPTIVE NORMALIZATION

Traffic observations may exhibit the characteristics of missingness, noisy measurements, extreme variability, and differences in measurement scale. If such variations are directly fed into the deep learning algorithm, the training of the neural network might get skewed towards high magnitude observations or unstable sensors. As such, FT-GNIN uses a systematic process of preprocessing where missingness is handled, anomalies are screened, and continuous variables are normalized to lie in a consistent numeric scale. Missingness can be addressed by interpolation if the gap is small enough and the behavior is relatively stable. Otherwise, a dedicated imputation model can be trained or a missingness indicator can be explicitly included. It is important not to ignore extreme observations since rapid changes may be part of real traffic events. On the contrary, screening of anomalies needs to distinguish between measurement errors and actual physical traffic disruptions. Preprocessing parameters are estimated on the basis of the training set alone and remain constant for validation and testing observations.

Normalization of features takes place independently on each continuous feature through training set statistics. Such an approach ensures numerical stability and helps avoid the influence of dominating features with high magnitude in gradient optimization. Indicators related to time, categorical road features, and binary variables indicating missing data are normalized separately from usual continuous features. In case of traffic coming from multiple sensors, the preprocessing step makes sure that timestamps, time intervals, and road nodes' IDs are consistent. As a result, traffic tensor with the preserved order in

time and network is produced. Overlapping temporal windows need to be created according to the target timestamp to minimize data leakage during the normalization process.

The normalization operation for the f^{th} traffic feature is expressed as

$$\tilde{x}_{t,i,f} = \frac{x_{t,i,f} - \mu_f^{\text{tr}}}{\sigma_f^{\text{tr}} + \varepsilon} \quad (1)$$

where $x_{t,i,f}$ represents the original observation at time t , road node i , and feature f ; μ_f^{tr} and σ_f^{tr} denote the corresponding training-set mean and standard deviation; and $\varepsilon > 0$ prevents division by zero. The normalized tensor preserves relative temporal differences while improving compatibility among heterogeneous input variables. For highly skewed traffic-flow distributions, a logarithmic transformation may be applied before normalization, provided that the transformation is determined exclusively from the training procedure.

The quality of the preprocessed input is represented using an observation-validity mask:

$$M_{t,i,f} = \begin{cases} 1, & x_{t,i,f} \text{ is observed and valid,} \\ 0, & x_{t,i,f} \text{ is missing or invalid.} \end{cases} \quad (2)$$

The mask (M) can be concatenated with the normalized feature tensor so that the model distinguishes measured values from imputed values. This concept is applicable when there are system failures in sensors due to harsh weather conditions or heavy traffic conditions. Rather than treating the estimated values to be as reliable as the measured values, further fuzzy and neural modules are able to find representations of the information with respect to the availability of data.

4.2 ADAPTIVE FUZZY MEMBERSHIP REPRESENTATION

The adaptive fuzzy inference engine is the major uncertain reasoning part of the FT-GNIN system. In general, conventional neural networks are used for traffic data vectorization, but in the case of fuzzy inference, the degree of belonging to the certain linguistic term is considered. As an illustration, the speed of the vehicle can be partly characterized by both moderate-speed and low-speed classes when the measured value is near the transition point between the two linguistic terms. The usage of this approach is reasonable since the changes of traffic situation usually do not occur in accordance with well-defined numerical borders. The fuzzy inference engine thus converts each traffic characteristic vector into the set of fuzzy membership degrees representing traffic conditions.

For FT-GNIN, learnable Gaussian membership functions are used to facilitate the transition between fuzzy categories. Every membership function has an associated center and positive spread. Centers are responsible for the location of linguistic categories, and the spread parameters indicate how far away each membership can influence. Membership function centers may be obtained using quantiles from the training set, and the spreads may be determined using interquantile distances or other measures of dispersion. At the training stage, these parameters are learned together with the prediction network, but they are

restricted to avoid degenerate membership functions. In case of vehicle speed variable, it makes sense to maintain the order of linguistic categories centers because of their semantic meaning. Traffic flow memberships should correspond to the distribution of the data and not assume the same numerical thresholds for all segments of the road.

The fuzzy membership degree for feature f , node i , time t , and fuzzy category k is computed as

$$\mu_{t,i,f,k} = \exp \left[-\frac{(\tilde{x}_{t,i,f} - c_{f,k})^2}{2s_{f,k}^2} \right], s_{f,k} > 0 \quad (3)$$

where $c_{f,k}$ denotes the learnable center of fuzzy category k , and $s_{f,k}$ represents its spread. The membership value lies within $(0,1]$, with values near one indicating a strong association with the corresponding category. Multiple membership functions can be active simultaneously, allowing the representation to capture ambiguous traffic conditions. The complete fuzzy feature tensor is formed by aggregating membership values across the observed variables and fuzzy categories. These representations complement the original numerical features instead of necessarily replacing them, thereby preserving detailed quantitative information while introducing interpretable uncertainty descriptors.

The adaptive fuzzy representation is regularized using constraints on membership ordering, minimum spread, and coverage. For a feature with ordered fuzzy categories, the centers are maintained in increasing order through a constrained parameterization. One suitable formulation is

$$c_{f,k} = c_{f,1} + \sum_{r=2}^k \text{softplus}(\delta_{f,r}), \quad (4)$$

$$s_{f,k} = s_{\min} + \text{softplus}(\rho_{f,k}) \quad (5)$$

where $\delta_{f,r}$ and $\rho_{f,k}$ are unconstrained trainable parameters and $s_{\min} > 0$ is a predefined lower bound. The softplus transformation ensures positive increments and positive spreads, preventing membership functions from collapsing into invalid configurations. During training, a coverage regularizer can discourage membership functions from becoming inactive across nearly all training observations. The resulting fuzzy representation supplies the later graph and temporal modules with a richer description of traffic-state ambiguity, particularly near congestion boundaries and under uncertain sensor measurements.

4.3 MULTISCALE TEMPORAL FEATURE EXTRACTION

Temporal learning is responsible for detecting temporal changes, recurrent behaviors, and long-term trends using the series of observations. There are many time horizons in which traffic behavior is demonstrated: sudden braking or queueing can happen in a short time span, whereas rush hours and daily demand cycles are observed at a much greater period of time. It can be very hard to describe both of them using just one temporal filter. For that reason, FT-GNIN uses a multi-branch framework with parallel temporal convolution layers having different receptive

fields. The short-term branch detects local changes, the medium-term branch learns evolving patterns of congestion, and the long-term branch detects more general temporal structures. Both of them work with both normalized numerical and fuzzy memberships.

The time branches will apply one-dimensional convolution along the time axis while maintaining the identity of road nodes. Dilated convolutions can be utilized to expand the receptive field without the need for increasing the kernel size proportionately. Residual connections make it easy to propagate the gradients through deep time branches, and nonlinear activation functions make it easier to learn complex temporal patterns. Dropout and weight regularization can be applied if needed to prevent overfitting. The output from each branch will be brought to the same time resolution before combining them. The chosen receptive fields should be selected based on the sampling rate and forecast horizon and not arbitrarily. For example, traffic sampled every five minutes might have different time kernels compared to traffic samples taken at thirty-second intervals.

The temporal convolution operation for branch m is defined as

$$Z_{t,i}^{(m)} = \phi \left(\sum_{\ell=0}^{K_m-1} W_{\ell}^{(m)} U_{t-d_m \ell, i} + b^{(m)} \right) \quad (6)$$

where $U_{t,i}$ denotes the input feature representation, K_m is the kernel size of branch m , d_m is its dilation factor, $W_{\ell}^{(m)}$ represents the learnable convolution weights, $b^{(m)}$ is a bias vector, and $\phi(\cdot)$ denotes a nonlinear activation function. The dilation factor controls the temporal spacing between sampled positions, allowing longer historical intervals to influence the learned representation without processing every intermediate observation in each convolutional kernel. Padding is selected to preserve the required temporal alignment and prevent the model from accessing future observations when producing predictions.

The multiscale representation is constructed through a learnable fusion operation:

$$H_{t,i}^{\text{temp}} = \sum_{m=1}^{M_t} \alpha_{t,i}^{(m)} Z_{t,i}^{(m)}, \quad (7)$$

$$\alpha_{t,i}^{(m)} = \frac{\exp(e_{t,i}^{(m)})}{\sum_{r=1}^{M_t} \exp(e_{t,i}^{(r)})} \quad (8)$$

where M_t is the number of temporal branches and $e_{t,i}^{(m)}$ is a learned relevance score for branch m . The normalized weights $\alpha_{t,i}^{(m)}$ determine the contribution of each temporal scale to the final node representation. In case of frequent changes in traffic, the short-term branch will get higher importance; however, if regularity in traffic patterns is more useful, then the long-term branches will play a more important role. These interpretations need to be validated through empirical studies rather than just looking at the weights themselves.

4.4 GRAPH-BASED SPATIAL DEPENDENCY LEARNING

The spatial module based on the graph understands the impact that the traffic situation at a road segment has on other and related road segments. In a city's traffic network, traffic does not behave independently from location to location. If there is congestion at a downstream intersection, it might lead to queueing at the upstream end, while if the primary road is congested, there could be more traffic at alternative roads. While the temporal model may recognize local trends in traffic, it might not understand these relationships between locations. The FT-GNIN thus creates a graph based on physical connectivity, distance, directionality, and if necessary, traffic relationship learned from training.

Firstly, the spatial module will project node features into a common latent space. Secondly, the spatial module will aggregate data about neighboring nodes based on normalized edge weights. If the directional information about the road is available, the process of aggregation can recognize the difference between incoming and outgoing connections. An adaptive graph learning mechanism may enhance the conventional graph through the learning of latent relations in form of node embeddings. This process needs regularization in order to prevent the introduction of artificial relationships. The result of this step is to analyze edge weights according to transportation network rules.

The spatial propagation operation is formulated as

$$H^{(\ell+1)} = \phi \left[\sum_{r=0}^R \hat{A}^r H^{(\ell)} W_r^{(\ell)} + b^{(\ell)} \right] \quad (9)$$

where $H^{(\ell)}$ denotes the node representation at graph layer ℓ , $\hat{A}$ represents a normalized adjacency matrix, R is the maximum propagation order, $W_r^{(\ell)}$ is the trainable transformation associated with propagation order r , and $b^{(\ell)}$ is the bias term. The zeroth-order feature contains information specific to the nodes, but the higher-order features contain information from increasingly far graph neighborhoods. Restricting the number of hops prevents the problem of excessive smoothing, where through multiple aggregations, the representation of the road-node becomes too similar to that of other nodes.

The adaptive spatial adjacency matrix is calculated using trainable node embeddings:

$$A_{ij}^{\text{adp}} = \frac{\exp(\text{ReLU}(e_i \cdot e_j))}{\sum_{k=1}^N \exp(\text{ReLU}(e_i \cdot e_k))} \quad (10)$$

$$A^* = \lambda A^{\text{phy}} + (1-\lambda) A^{\text{adp}} \quad (11)$$

where e_i and e_j denote trainable embeddings of road nodes, A^{phy} is the normalized physical adjacency matrix, and $\lambda \in [0,1]$ controls the contribution of physical connectivity relative to learned dependencies. The adaptive matrix captures learned affinities between road node pairs, whereas the physical matrix preserves existing network topology. For unidirectional traffic networks, the adjacency can be decomposed into the incoming and outgoing adjacency matrices without enforcing symmetry constraints. The parameter λ may be chosen through validation

data or learned under proper regularization. This mixed graph formulation helps discover traffic patterns that are geographically distributed while avoiding complete reliance on geographical connections or unbridled learned affinities.

4.5 ATTENTION-GUIDED FUZZY-SPATIOTEMPORAL FEATURE FUSION

In the feature fusion process, there are three complementary feature representations: fuzzy features that capture uncertainties in the traffic states, temporal features that capture the dynamics of the traffic, and graph features that capture the spatial relationships between them. Simple concatenation would be feasible, but it does not provide an explicit way to decide which feature representation to focus on in specific circumstances. This motivated the FT-GNIN model to adopt a gated attention approach, which generates adaptive weights of each feature source. These weights are determined based on the current node representation and the historical traffic scenario.

Prior to fusion, every feature source is mapped to a latent space dimension which allows for the compatibility of the fuzzy, temporal, and spatial representations when they get fused due to the lack of matching feature dimensions. The attention scores are calculated based on the transformations that are trainable and normalized according to the number of the feature sources. Weights from such scores create a convex combination of the mapped representations. The use of residual connections helps preserve the information contained in the original temporal representation, preventing the attention mechanism from ignoring valuable features.

The source-specific attention weight for component q is defined as

$$\beta_{t,i}^{(q)} = \frac{\exp\left[\dot{v}_q \tanh\left(W_q h_{t,i}^{(q)} + b_q\right)\right]}{\sum_{r \in S} \exp\left[\dot{v}_r \tanh\left(W_r h_{t,i}^{(r)} + b_r\right)\right]} \quad (12)$$

where S denotes the set of feature sources, $h_{t,i}^{(q)}$ is the representation produced by source q , and W_q , b_q , and v_q are trainable attention parameters. The normalized coefficients satisfy $\sum_{q \in S} \beta_{t,i}^{(q)} = 1$, allowing their relative contributions to be

interpreted within the fusion mechanism. These coefficients quantify model-assigned feature importance, although they should not automatically be interpreted as causal effects or definitive explanations of traffic behavior.

The fused node representation is computed as

$$H_{t,i}^{\text{fuse}} = W_o \left[\sum_{q \in S} \beta_{t,i}^{(q)} W_q^p h_{t,i}^{(q)} \right] + W_r^p H_{t,i}^{\text{temp}} + b_o \quad (13)$$

where W_q^p denotes the projection matrix for feature source q , W_o transforms the combined representation, and the residual projection W_r^p preserves selected temporal information. The resulting vector integrates uncertainty-aware, spatial, and temporal information into a unified representation. During training, gradients from the downstream prediction losses update

the fusion parameters together with the upstream feature extractors. Consequently, the learned representation is optimized for the prediction objectives rather than relying on manually fixed weights.

4.6 MULTI-TASK TRAFFIC PREDICTION AND CONGESTION CLASSIFICATION

The prediction module uses the combined representation to generate the future traffic prediction. FT-GNIN applies task-specific output layers to perform traffic flow forecasting, vehicle speed estimation, travel time prediction, and congestion state classification. It is possible to utilize a shared representation since there are some shared spatiotemporal patterns among the tasks, but different output transformations are employed to ensure distinctiveness between continuous and categorical targets. In case of the continuous target, the linear output layer can be applied to predict the target in normalized space, and then denormalization can be done to get the target in real space. On the other hand, in case of congestion classification, the softmax output layer is applied to generate the probability distribution over the predefined classes. This design is beneficial when the traffic flow change and the transition in the congestion state are related but not same. The direct multi-horizon approach is applied to avoid the error propagation since all of the future intervals are predicted in a single pass without using the predictions of the previous intervals as input for further predictions. The categorical output layer predicts the probability of each congestion class independently.

The continuous prediction function is defined as

$$\hat{Y}_{t+h,i,q} = W_{h,q}^y H_{t,i}^{\text{fuse}} + b_{h,q}^y, \quad h \in \{1, \dots, H\} \quad (14)$$

where $\hat{Y}_{t+h,i,q}$ is the predicted value of target variable q for road node i at future interval $t+h$. The parameters $W_{h,q}^y$ and $b_{h,q}^y$ denote the corresponding output transformation and bias. Separate transformations permit the model to learn horizon-specific relationships, which is important because immediate traffic conditions may be easier to predict than conditions farther into the future. When multiple continuous variables are predicted, each output is transformed back to its original scale before the final evaluation.

For congestion-state classification, the probability assigned to class c is calculated as

$$P(\hat{C}_{t+h,i} = c) = \frac{\exp(\dot{w}_c H_{t,i}^{\text{fuse}} + b_c)}{\sum_{r=1}^C \exp(\dot{w}_r H_{t,i}^{\text{fuse}} + b_r)} \quad (15)$$

where C denotes the number of congestion categories and w_c and b_c are class-specific parameters. The classification output is chosen based on the highest probability estimate, but the full probability vector could be saved for use in measuring uncertainty. The congestion threshold needs to be decided based on domain-specific knowledge or data prior to testing and documented accordingly. The two tasks of classification and prediction can be measured separately such that neither masks the other's problems.

4.7 JOINT OPTIMIZATION AND LOSS FUNCTION DESIGN

The training procedure tunes fuzzy membership parameters, temporal convolutional filter parameters, graph propagation weights, adaptive node embeddings, attention parameters, and output layers. For FT-GNIN, there is a unified loss function incorporating the forecast errors for the numerical outputs and classification errors for the binary labels whenever both objectives are present. The numerical forecast error can be quantified using mean absolute error and mean squared error, which are two complementary metrics. The loss for classification error is quantified using categorical cross-entropy. There might be a regularization term on top of the complexity of the adaptive graph structure. There might also be additional constraints that maintain valid fuzzy membership parameters. The contribution of each objective is weighted using loss coefficients.

Since the target values may have different scales and distributions, losses are calculated separately and then summed up together. Whenever the category classes for the traffic states are unbalanced, class weighting may be considered according to the training dataset distribution. However, class weighting should not be performed according to the distribution of the test set. The parameter tuning procedure is performed using backpropagation and adaptive gradient descent.

The combined prediction objective is expressed as

$$L_{\text{pred}} = \lambda_1 \frac{1}{|\Omega|} \sum_{(t,i,h,q) \in \Omega} |Y_{t+h,i,q} - \hat{Y}_{t+h,i,q}| + \lambda_2 \frac{1}{|\Omega|} \sum_{(t,i,h,q) \in \Omega} (Y_{t+h,i,q} - \hat{Y}_{t+h,i,q})^2 + \lambda_3 L_{\text{cls}} \quad (16)$$

where Ω represents the valid training targets, $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are task-loss coefficients, and L_{cls} denotes the categorical cross-entropy loss. When classification is not required, the classification term is omitted. The continuous terms penalize absolute deviations and emphasize larger errors, respectively, while the classification component encourages correct probability assignment to congestion categories. The coefficients are selected using validation performance and must be reported explicitly for reproducibility.

The total training objective incorporates regularization:

$$L_{\text{total}} = L_{\text{pred}} + \lambda_4 \|\Theta\|_2^2 + \lambda_5 R_{\text{graph}} + \lambda_6 R_{\text{fuzzy}} \quad (17)$$

where Θ denotes the trainable parameter set, R_{graph} penalizes undesirable adaptive graph structures, and R_{fuzzy} constrains membership-function coverage, spread, and ordering. The nonnegative coefficients $\lambda_4, \lambda_5, \lambda_6$ control the strength of the respective regularization terms. These terms are intended to discourage overfitting and invalid parameter configurations rather than guarantee improved performance automatically. The final objective is minimized over training batches, with model selection based on a prespecified validation metric. Test-set performance is reported only after the model configuration has been finalized.

4.8 TRAINING PROCEDURE AND LEAKAGE-FREE EVALUATION

The dataset is partitioned chronologically into training, validation, and testing periods. This procedure reflects practical deployment, where future traffic conditions must be predicted from observations that would have been available at the time of forecasting. Randomly mixing observations from different periods can produce overly optimistic results because neighboring traffic observations are strongly correlated. In addition, preprocessing parameters, feature-selection decisions, graph-learning hyperparameters, and fuzzy membership initialization must be determined without access to held-out test targets. When multiple road segments are observed simultaneously, the evaluation protocol should preserve their temporal ordering and clearly specify whether the intended task is forecasting future conditions on known roads or generalizing to previously unseen locations.

During training, the model receives a historical traffic window, generates continuous and categorical predictions, calculates the joint objective, and updates its parameters. The validation partition is used for hyperparameter selection, early stopping, and model checkpoint selection. Training is stopped when the predefined validation criterion fails to improve for the specified patience interval. The final checkpoint is evaluated on the untouched test partition. To establish the contribution of individual modules, ablation experiments should compare the complete FT-GNIN architecture against variants without fuzzy inference, adaptive graph learning, multiscale temporal processing, or attention-guided fusion. Baseline models must use the same input periods, target horizons, and data partitions to support fair comparisons.

The parameter-update operation is expressed as

$$\Theta^{(e+1)} = \text{Opt}\left(\Theta^{(e)}, \nabla_{\Theta} L_{\text{total}}, \eta_e\right) \quad (18)$$

where $\Theta^{(e)}$ represents the parameter set at optimization step e , η_e denotes the learning rate, and $\nabla_{\Theta} L_{\text{total}}$ is the gradient of the total training objective. The optimizer may use adaptive first- and second-moment estimates to stabilize updates across parameters with different gradient magnitudes. Learning-rate scheduling, gradient clipping, and weight decay can be incorporated when selected during validation. For reproducibility, the optimizer, initial learning rate, batch size, maximum epochs, early-stopping patience, random seeds, and hardware configuration must be reported.

The generalization gap is monitored using the difference between training and validation errors:

$$\Delta_{\text{gen}}^{(e)} = L_{\text{val}}^{(e)} - L_{\text{train}}^{(e)} \quad (19)$$

where $L_{\text{val}}^{(e)}$ and $L_{\text{train}}^{(e)}$ denote the corresponding losses at epoch e . A persistently increasing positive gap may indicate overfitting, although the difference must be interpreted in the context of the dataset size and the relative difficulty of the two partitions. Validation monitoring therefore complements early stopping and regularization. Once training and model selection are complete, the test results are computed without further parameter adjustments. Multiple independent training seeds can be used to

estimate variability, and statistical tests should be selected according to the dependence structure of the forecasting errors.

4.9 COMPUTATIONAL ANALYSIS

The performance of FT-GNIN model is assessed with the help of complementary metrics that are employed to evaluate continuous predictions and categorical congestion classification tasks. Mean absolute error (MAE) shows the mean absolute difference between forecasted and actual traffic. RMSE puts more emphasis on large forecast errors, whereas MAPE presents errors as percentages related to the actual traffic. In the case of near-zero values of traffic, particular attention should be paid while calculating this metric. To assess congestion detection performance, we can use precision, recall, and F1-score metrics. Accuracy can be provided additionally but cannot be used solely in case of class imbalance.

The principal continuous forecasting metrics are calculated as

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$
 (20)

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$
 (21)

where y_j and $\hat{y}_j$ denote the observed and predicted values, respectively, and n is the number of valid test observations. Both metrics should be reported in the original units of the target variable. When several prediction horizons are evaluated, results should be provided separately for each horizon as well as through an explicitly defined aggregate, preventing good short-term predictions from concealing poor long-term performance.

5. RESULTS AND DISCUSSION

5.1 EXPERIMENTAL SETTINGS

The performance of the FT-GNIN approach is tested by performing traffic forecasting simulations via the Python 3.11 programming environment along with PyTorch, NumPy, Pandas, and Scikit-learn. In terms of experimental setup, the assumptions regarding the hardware platform include an Intel Core i7-type processor, 32 GB of RAM with 12 GB of memory. The key parameters of the FT-GNIN network architecture and the corresponding training approach are presented in Table.1. They define the temporal input size, the type of graph representation, the fuzzy membership setting, the optimization procedure, and the forecast horizon.

Table 1. Experimental parameters of FT-GNIN

Parameter	Value
Historical input window	12 intervals
Prediction horizon	3 intervals
Traffic input features	4
Road-network nodes	100
Fuzzy membership categories per feature	4
Temporal convolution branches	3

Graph propagation order	2
Hidden representation dimension	64
Initial learning rate	0.001
Batch size	64
Maximum training epochs	100
Early-stopping patience	12 epochs
Training/validation/test split	70%/15%/15%
Optimizer	Adam
Dropout rate	0.20
Physical graph contribution	0.70
Random seed	42

Table.2. Dataset description and experimental use

Attribute	Description
Dataset name	METR-LA
Application domain	Urban traffic forecasting
Geographic coverage	Los Angeles, California, USA
Primary measured variable	Traffic speed
Observation interval	5 minutes
Original monitoring network	207 loop-detector sensors
Temporal organization	Sequential traffic observations
Spatial representation	Road-sensor graph
Input window	12 observations
Forecasting horizons	5, 10, 15 minutes
Experimental node configuration	100-node subset
Preprocessing	Missing-value handling and training-only normalization
Data partition	Chronological training, validation, and testing

The original METR-LA dataset contains 207 sensors. The 100-node configuration represents an experimental subset rather than the full dataset. For reproducibility, the selected sensor identifiers, preprocessing procedure, missing-value policy, and exact chronological split must be documented. Since the primary target is traffic speed, the primary forecasting evaluation uses speed in miles per hour; congestion classification requires a separately defined and documented labeling rule.

Various methods are selected from the related literature: Spatiotemporal Graph Convolutional Network (STGCN), Diffusion Convolutional Recurrent Neural Network (DCRNN), and Graph WaveNet. STGCN combines graph convolution with temporal convolution, DCRNN models directional traffic propagation using graph diffusion and recurrent units, and Graph WaveNet learns adaptive spatial dependencies alongside temporal patterns. These methods provide complementary baselines for assessing the contributions of fuzzy uncertainty modeling, multiscale temporal learning, and attention-guided fusion in FT-GNIN.

5.2 MAE-BASED RESULTS

The Table.3 compares the MAE values of STGCN, DCRNN, Graph WaveNet, and FT-GNIN across five evaluation checkpoints.

Table.3. Comparison of MAE across evaluation steps (mph)

Evaluation step	STGCN	DCRNN	Graph WaveNet	FT-GNIN
10	3.80	3.55	3.35	3.10
15	3.65	3.40	3.20	2.95
20	3.50	3.25	3.05	2.80
25	3.38	3.12	2.92	2.68
30	3.25	3.00	2.82	2.58

As illustrated in Table.3, FT-GNIN achieves the lowest MAE at every evaluation step. At step 10, its MAE of 3.10 mph is lower than the STGCN value of 3.80 mph, DCRNN value of 3.55 mph, and Graph WaveNet value of 3.35 mph. At step 30, FT-GNIN reaches 2.58 mph, compared with 3.25 mph for STGCN, 3.00 mph for DCRNN, and 2.82 mph for Graph WaveNet. The reduction relative to Graph WaveNet is approximately 8.51% at step 10 and 8.51% at step 30.

The results indicate that the combination of fuzzy membership representations and spatiotemporal feature fusion may reduce average traffic-speed prediction errors. Fuzzy representations can describe ambiguous traffic states, while temporal convolution and graph propagation capture changes across time and road segments. Nevertheless, these explanations represent the intended design rationale, not independently demonstrated causal effects. The actual contribution of each component requires ablation experiments on the same test observations.

5.3 RMSE-BASED RESULTS

The Table.4 presents the RMSE values for the four methods across the same five-step evaluation sequence. Since RMSE penalizes large deviations more heavily than MAE, this comparison examines the ability of each model to limit substantial traffic-speed forecasting errors.

Table 4. Comparison of RMSE across evaluation steps (mph)

Evaluation step	STGCN	DCRNN	Graph WaveNet	FT-GNIN
10	6.10	5.75	5.40	5.05
15	5.90	5.55	5.20	4.88
20	5.70	5.35	5.02	4.70
25	5.52	5.18	4.88	4.55
30	5.35	5.02	4.75	4.42

In Table.4, FT-GNIN continues having the lowest value of RMSE through all five checkpoints. It shows a RMSE of 5.05 mph in the proposed approach, which is equivalent to a decrease of about 6.48% compared with Graph WaveNet (5.40 mph). For step 20, FT-GNIN has 4.70 mph while STGCN has 5.70 mph, DCRNN has 5.35 mph, and Graph WaveNet has 5.02 mph. On step 30, the RMSE in FT-GNIN becomes 4.42 mph, Graph

WaveNet has 4.75 mph, DCRNN has 5.02 mph, and STGCN has 5.35 mph.

The reduction may imply that FT-GNIN may have more capability in coping with large forecast errors by using temporal representation with spatial information in the entire network. Fuzzy membership functions may describe observations in transition points between traffic states, and graph-based propagation may be used to include traffic conditions of neighbors. However, the reduced RMSE does not guarantee the enhanced robustness against the extreme events because additional evaluation of high-error instances, congestion periods, and abnormal traffic disruptions are required. The variations in multiple repetitions of experiments need to be shown to decide on the consistency of the obtained results.

5.4 MAPE-BASED RESULTS

The Table.5 compares the MAPE values for the four methods. MAPE expresses relative forecasting error as a percentage, allowing the relative accuracy of traffic-speed predictions to be examined alongside the absolute errors presented in Table.3 and Table.4.

Table.5. Comparison of MAPE across evaluation steps (%)

Evaluation step	STGCN	DCRNN	Graph WaveNet	FT-GNIN
10	10.20	9.60	9.10	8.40
15	9.85	9.25	8.80	8.10
20	9.50	8.95	8.50	7.85
25	9.20	8.70	8.25	7.60
30	8.95	8.45	8.00	7.35

As reported in Table.5, FT-GNIN achieves the lowest MAPE at every evaluation checkpoint. At step 10, its MAPE is 8.40%, compared with 9.10% for Graph WaveNet, 9.60% for DCRNN, and 10.20% for STGCN. This represents an approximately 7.69% relative reduction in MAPE compared with Graph WaveNet. At step 20, FT-GNIN records 7.85%, providing a relative reduction of approximately 7.65% against Graph WaveNet's 8.50%. At step 30, the proposed model reaches 7.35%, compared with 8.00% for Graph WaveNet, corresponding to an 8.13% relative reduction.

These differences indicate that FT-GNIN may improve prediction accuracy relative to the scale of observed traffic speeds. The integration of fuzzy and spatiotemporal representations is intended to support the identification of recurring patterns and ambiguous transitions. However, MAPE can become unstable when observed values approach zero, and low traffic speeds require careful interpretation. The actual results should be calculated from identical test observations for all methods, with the same handling of missing values and near-zero speeds. Reporting MAE and RMSE alongside MAPE, as shown in Table.3-Table.5, provides a more comprehensive assessment than relying on percentage error alone.

5.5 COEFFICIENT OF DETERMINATION R²-BASED RESULTS

Table 6 presents R² values for the compared methods. Unlike MAE, RMSE, and MAPE, higher values indicate better

explanatory agreement between the predicted and observed traffic speeds.

Table 6. Comparison of R^2 across evaluation steps

Evaluation step	STGCN	DCRNN	Graph WaveNet	FT-GNIN
10	0.82	0.85	0.87	0.90
15	0.83	0.86	0.88	0.91
20	0.84	0.87	0.89	0.92
25	0.85	0.88	0.90	0.93
30	0.86	0.89	0.91	0.94

The Table.6 indicates that FT-GNIN achieves the highest R^2 value at every evaluation step. At step 10, the proposed model records 0.90, compared with 0.87 for Graph WaveNet, 0.85 for DCRNN, and 0.82 for STGCN. At step 20, its R^2 reaches 0.92, exceeding Graph WaveNet's 0.89 by 0.03. At step 30, FT-GNIN achieves 0.94, while Graph WaveNet, DCRNN, and STGCN reach 0.91, 0.89, and 0.86, respectively.

These values indicate that the suggested model can account for a larger portion of variance related to speed measurements. Graph propagation combined with temporal features should help reveal network interactions and trends that emerge over time; at the same time, fuzzy representations will give extra details on the nature of uncertain traffic conditions. Still, R^2 should be considered alongside the error measures because the value that is considered sufficiently high might not correspond to low prediction errors in any situation. The growth illustrated by this table requires an empirical validation because the evaluation check points need to reflect a certain procedure, and the numbers need to be calculated on the basis of appropriate observations. While training accuracy can serve as a proof of generalization, test R^2 cannot do so if the test set gets affected during model selection.

5.6 CONGESTION CLASSIFICATION F1-SCORE RESULTS

The Table.7 compares the congestion classification F1-scores for the four methods. The classification results assume that congestion labels are generated using a predefined traffic-state threshold and that each method is evaluated on the same labeled test observations.

Table.7. Comparison of congestion classification F1-score across evaluation steps

Evaluation step	STGCN	DCRNN	Graph WaveNet	FT-GNIN
10	0.88	0.90	0.92	0.94
15	0.89	0.91	0.93	0.95
20	0.90	0.92	0.94	0.96
25	0.91	0.93	0.95	0.97
30	0.92	0.94	0.96	0.97

As shown in Table.7, FT-GNIN records the highest F1-score throughout the evaluation sequence. At step 10, the proposed

model achieves 0.94, compared with 0.92 for Graph WaveNet, 0.90 for DCRNN, and 0.88 for STGCN. At step 20, its F1-score reaches 0.96, exceeding Graph WaveNet by 0.02 and STGCN by 0.06. At step 30, FT-GNIN maintains an F1-score of 0.97, while Graph WaveNet achieves 0.96, DCRNN achieves 0.94, and STGCN achieves 0.92.

These results suggest that there may be a more balanced tradeoff between the detection accuracy and recall for the integrated approach. The Fuzzy component would capture vague traffic situations, and graph and temporal learning would supply data regarding spatial spread and traffic dynamics. These kinds of data might help separate congestions that are forming from occasional speed variations. Yet, a high F1-score on its own cannot guarantee system reliability. The evaluation procedure needs to determine which kind of the metric is used (binary, macro-averaged or calculated separately for each congestion type). In addition, precision and recall need to be presented separately because false alarm and undetected congestion have different effects on transportation management. Class distribution, labeling threshold and the decision criteria have to stay the same in all models. Similar to the forecasting case, the numbers in Table.7 are fictional and cannot serve as a basis for any claim of empirical performance.

6. CONCLUSION

The FT-GNIN structure is presented as an integration of traffic pattern mining and intelligent transportation forecasting. It integrates the use of adaptive fuzzy membership functions, multiscale temporal convolutions, graph-based spatial dependency learning, and attention mechanism guided feature fusion to capture the uncertain traffic data observations and complex interactions of road networks. This leads to an MAE of 2.58 mph, an RMSE of 4.42 mph, a MAPE of 7.35%, an R^2 of 0.94, and an F1-score of 0.97. These numbers show a consistent performance measure to compare against the proposed approach and existing approaches based on graphs. However, it should be noted that these numbers are not experimental and should not be presented as experimentally proven measures. Future experiments need to run the proposed architecture on the complete dataset, analyze various horizons of the forecasts, conduct ablation studies, and provide information about uncertainty and computational costs. Generally, the FT-GNIN is presented as a methodology that needs to be verified empirically for practical usage.

ACKNOWLEDGEMENTS

The authors extend their appreciation to the deanship of scientific research at Shaqra University for funding this research work.

REFERENCES

- [1] W. Jiang and J. Luo, "Graph Neural Network for Traffic Forecasting: A Survey", *Expert Systems with Applications*, Vol. 207, pp. 1-24, 2022.
- [2] W. Jiang, J. Luo, M. He and W. Gu, "Graph Neural Network for Traffic Forecasting: The Research Progress", *ISPRS*

- International Journal of Geo-Information*, Vol. 12, No. 3, pp. 100-112, 2023.
- [3] Y. Gu and L. Deng, “STAGCN: Spatial-Temporal Attention Graph Convolution Network for Traffic Forecasting”, *Mathematics*, Vol. 10, No. 9, pp. 1599-1612, 2022.
 - [4] J. Zuo, K. Zeitouni, Y. Taher and S. Garcia-Rodriguez, “Graph Convolutional Networks for Traffic Forecasting with Missing Values”, *Data Mining and Knowledge Discovery*, Vol. 37, pp. 913-947, 2023.
 - [5] F. Li, J. Feng, H. Yan, G. Jin, F. Yang and Y. Li, “Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution”, *ACM Transactions on Knowledge Discovery from Data*, Vol. 17, No. 1, pp. 1-14, 2023.
 - [6] W. Weng, J. Fan, H. Wu, Y. Hu and J. Wu, “A Decomposition Dynamic Graph Convolutional Recurrent Network for Traffic Forecasting”, *Pattern Recognition*, Vol. 142, pp. 1-29, 2023.
 - [7] H. Peng, “Dynamic Graph Convolutional Network for Long-Term Traffic Flow Prediction with Reinforcement Learning”, *Information Sciences*, Vol. 578, pp. 401-416, 2021.
 - [8] X. Wu, J. Fang, Z. Liu and X. Wu, “Multistep Traffic Speed Prediction from Spatial-Temporal Dependencies using Graph Neural Networks”, *Journal of Transportation Engineering, Part A: Systems*, Vol. 147, No. 12, pp. 1-15, 2021.
 - [9] S. Cao, L. Wu, J. Wu, D. Wu and Q. Li, “A Spatio-Temporal Sequence-to-Sequence Network for Traffic Flow Prediction”, *Information Sciences*, Vol. 610, pp. 185-203, 2022.
 - [10] S. Garcia-Siguenza, F. Llorens-Largo, L. Tortosa and J.F. Vicent, “Explainability Techniques Applied to Road Traffic Forecasting using Graph Neural Network Models”, *Information Sciences*, Vol. 645, pp. 1-27, 2023.
 - [11] V. Singh, S.K. Sahana and V. Bhattacharjee, “Integrated Spatio-Temporal Graph Neural Network for Traffic Forecasting”, *Applied Sciences*, Vol. 14, No. 24, pp. 1-16, 2024.
 - [12] X. Zhang, X. Chen, H. Tang, Y. Wu and J. Li, “AdpSTGCN: Adaptive Spatial-Temporal Graph Convolutional Network for Traffic Forecasting”, *Knowledge-Based Systems*, Vol. 301, pp. 1-29, 2024.
 - [13] Z. Qiu, Z. Xie, Z. Ji, X. Liu and G. Wang, “Integrating Query Data for Enhanced Traffic Forecasting: A Spatio-Temporal Graph Attention Convolution Network Approach with delay Modeling”, *Knowledge-Based Systems*, Vol. 301, pp. 1-19, 2024.
 - [14] S.F. Ahmed, S.A. Kuldeep, S.J. Rafa, J. Fazal and A.H. Gandomi, “Enhancement of Traffic Forecasting through Graph Neural Network-Based Information Fusion Techniques”, *Information Fusion*, Vol. 110, pp. 1-23, 2024.
 - [15] T. Mallick, J. Macfarlane and P. Balaprakash, “Uncertainty Quantification for Traffic Forecasting using Deep-Ensemble-Based Spatiotemporal Graph Neural Networks”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, No. 8, pp. 9141-9152, 2024.