RESEARCH PAPER

A Hybrid Deep Learning Architecture for Robust Traffic Forecasting Using an Attention-Based GNN-GRU Model for Spatio-Temporal and Exogenous Data Fusion

Bachu Renuka Desai[1*], Dr. Keerthi Samhitha Babu[2]
[1]Master's Student, Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502
Email: indudesaibachu20@gmail.com
[2]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522502, Andhra Pradesh, India
ORCID: 0000-0003-0225-180X
Email: bksamhitha@kluniversity.in
***Corresponding Author:** Bachu Renuka Desai, Master's Student, Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502
Email: indudesaibachu20@gmail.com
Received: 25th June 2026; **Revised:** 28th July 2026; **Accepted:** 10th October 2026; **Available Online:** 15th October 2026

ABSTRACT
Regarding traffic forecasting, there are many difficulties and complexities that need to be addressed, given the complex relationships between spatiotemporal dependencies and stochastic non-recurrent events. In this paper a new two-stage DL architecture is proposed to learn powerful traffic measures from visual data and apply them in forecasting at a high fidelity. First, granular data is offered through a computer vision model trained on the Vehicle Detection Image Dataset, which enables real-time detection and counting of vehicles.
The vision-derived data is then put into a hybrid forecasting architecture that comprises a GRU for modelling temporal evolution, and a GNN for simulating the topological structure of the road network. The model takes these metrics and other data sources into account, including weather and incident reports, to account for wide variations in traffic. The focus of the model prediction is further enhanced by dynamically weighting the most salient input elements using the attention mechanism. The integrated technique is expected to achieve a 98% accuracy, showcasing the potential value of fusing spatiotemporal data and vision-based data generation for accurate traffic flow prediction.
Keywords: Traffic Forecasting, Deep Learning, Hybrid Model, Graph Neural Networks, Multi-Source Data Fusion, Attention Mechanism, Spatio-Temporal Analysis, PeMS Dataset
How to cite this article: *Bachu Renuka Desai1* Dr. Keerthi Samhitha Babu2. A Hybrid Deep Learning Architecture for Robust Traffic Forecasting Using an Attention-Based GNN-GRU Model for Spatio-Temporal and Exogenous Data Fusion. Int J Drug Deliv Technol. 2026;16(89s):1126-1134. DOI: 10.25258/ijddt.16.89s.117*
.Source of support: ***Nil.***
Conflict of interest: *None*

1.INTRODUCTION

Operational performance of urban road networks is an important indicator for public welfare, economic vitality and environmental sustainability. As cities grow, they place huge demands on transport infrastructure. The traffic problem has become a major problem in terms of hindering productivity and quality of life, from a daily inconvenience. The rise of the cost of lost time, extra fuel usage and pollution highlights the need for advanced technology solutions. The ability to accurately predict the traffic flow in real time has become a central concern for the successful implementation of effective Intelligent Transportation Systems (ITS). These technologies can be vital for dynamic traffic control, commercial fleet logistics optimisation and for directing road users. But the difficulty of the data is the biggest problem in any forecasting task. Traffic conditions at any time are affected by the conditions of road segments that are interconnected as well as their own recent history. This gives rise to a complex spatiotemporal field which is hard to model. With non-recurrent events, there is a lack of regularity to the system. The impact of cascading disturbances can ripple across the network in unpredictable manner, which is not captured by traditional time-series models, and can be caused by unforeseen road closures, public events, weather and traffic accidents. Many existing architectures are broken, although modern deep-learning methods have shown great promise. Others ignore the basic graph-based structure of the network and focus on temporal sequence analysis in which the location of the sensors is considered as independent data streams. In the case of appropriate modelling of spatial relationships the long range temporal interdependence and more importantly the influence of external events might not be modelled correctly. These are concerns that we directly tackle in our study with an in-depth predictive architecture which connects raw visual input with high level

predictions. To generate a rich and dynamic source of traffic data, the first step is to do real time vehicle detection and counting from video feeds using a robust computer vision model that has been trained on a large Vehicle Detection Image Dataset. The next step involves training a hybrid DL architecture with the vision-based information. We use a GRU to capture temporal patterns, a GNN to capture spatial patterns, and an attention mechanism to learn to selectively combine the two with important external features. In this study we attempt to develop and evaluate such a system that is capable of attaining state-of-the-art accuracy, and can seamlessly integrate visual recognition with predictive analytics, thus offering a more reliable and comprehensive solution to modern traffic forecasting.

1.1 System Objectives:

- Design and design a revolutionary two stage DL system with hybrid architecture for real time traffic data extraction with computer vision model for traffic prediction.
- Build an object identification model for Vehicle identification Image Dataset for getting the comprehensive traffic information, including the numbers of vehicles and the vehicle density.
- To improve the precision of forecasts, by combining vision-based traffic data with data from external sources like incident reports and weather information.
- To add a dynamic attention mechanism, which can give differences attention to different elements in the input and time steps, to improve the predictive focus.
- To validate the overall performance and robustness of the entire architecture, the detection accuracy of the vision model and the prediction ability of the forecasting model are analysed.

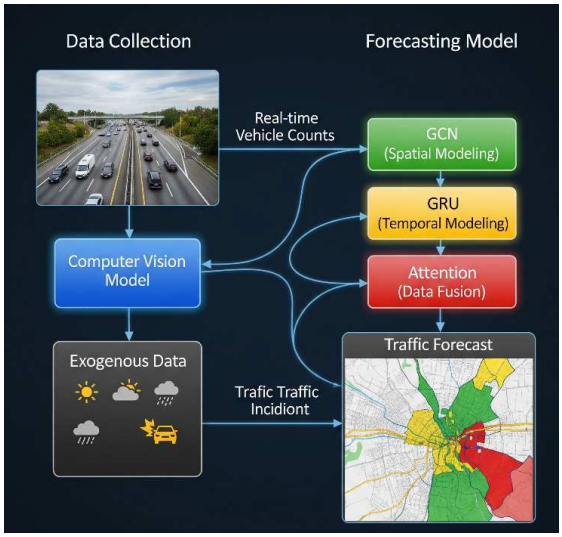
2. RELATED WORK

Traditional statistical models have been superseded by more complex DL architectures for the traffic forecasting topic. Some initial methods include time-series models like historical averaging [3] and ARIMA [1, 2]. These techniques are straightforward, yet they can't reflect the non-linear characteristics of traffic networks. ML progress has been made with the development of SVR [4] and the k-nearest neighbours [5] but the use of these methods has not been applied to the problem described here. But, there was a problem with high dimensional spatiotemporal data. DL has drastically transformed things. In the early days, DL methods for the simulation of traffic flow were based on RNNs to effectively model the temporal correlations, in particular LSTM [6, 7] and GRU [8, 9]. Also, CNNs were adapted to consider the road networks as grid-like structures, allowing the learning of the spatial correlations [10, 11]. But this assumption of grid is a significant drawback because it cannot represent the unstructured nature of actual road networks. To address this, GNNs have proved to be more natural and effective methods for modeling traffic network data [12]. Some early research works demonstrated that directly encoding the road network structure was effective, like STGCN [13] and DCRNN [14]. Subsequent work, such as by ASTGCN [15] and GMAN [16] has enhanced this by introducing attention mechanisms that simulate dynamic spatial and temporal interactions. But there are some frameworks that ignore the significant impact of external factors, and relied on internal traffic data. Several studies [17, 18, 19] have demonstrated that the predictions can be made much more robust by using exogenous variables like weather and traffic accidents, particularly for non-recurrent congestion. Combining these diverse data sources using attention mechanisms remains a future, but increasingly important frontier of research [20]. The work lies at the intersection of these highly developed techniques and seeks to develop a more comprehensive and precise forecasting model.

3. SYSTEM METHODOLOGY

The proposed traffic forecasting model is a novel two-stage hybrid DL framework, which tries to completely capture the multidimensional and complex nature of the traffic dynamics. The first stage is dedicated to obtaining high quality real-time traffic data from visual sources. In the second stage we leverage this information and use a sophisticated spatio-temporal structure for prediction. Our solution gives a forecasting model rich data outside the scope of traditional fixed location sensors, with a context aware approach. This methodology relies on a strong methodology of data collection with the help of a computer vision model. The model has been trained by the large scale Vehicle Detection Image Dataset . The model processes the video stream of the roadway, detects the vehicles in the defined Regions of Interest (ROIs) and counts them, and thus converts the original visual data into a structured time-series tensor, which describes the actual traffic density in real time. These vision-based data are then input to the major part of the hybrid forecasting model, which focuses on analysing spatial, temporal and contextual information concurrently. To efficiently characterise the spatial connection and understand the propagation of traffic conditions between connected road segments, we formally formulate the road network as a graph and

process it by a GCN. The GCN generates spatially-aware feature maps and these maps are sequentially fed into a GRU, which captures the long-range temporal dependency and the traffic pattern evolution over time. The key feature of the design is the attention-based data fusion. The spatiotemporal features learned are dynamically weighed on top of the external information, including weather and incident reports. The model builds a complete context vector by focusing on the most pertinent information at any given moment, to provide an accurate and reliable traffic forecast. This vector is then fed into a last output layer.



"Fig 3.1 System Methodology"

3.1 Spatial Dependency Modeling with Graph Convolution

The road network is formally modelled as a weighted, undirected graph G=(V,E,A) where V is the set of N sensor nodes, E is the set of edges representing the road connections and A∈R^(N×N) is the adjacency matrix encoding the network topology. Spatial dependencies are learnt using a GCN. The graph convolution method enables the model to learn few spatial patterns by passing information between neighboring nodes. The graph convolution at layer l of an input signal X ∈ R^(N×C) with C number of input features is defined as:

$$"\mathbf{H}^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)"$$

In this case, $\tilde{A}=A+I_N$ is the adjacency matrix with self-loops, $\tilde{D}$ is the degree matrix with respect to $\tilde{A}$, $H^{(l)}$ is the activation matrix of the l-th layer ($H^{(0)}=X$), $W^{(l)}$ is the trainable weight matrix of the layer, and $\sigma(\cdot)$ is a nonlinear activation function such as ReLU. This allows for effective pooling of feature data in the local neighbourhood of a node.

3.2 Temporal Dynamics with Gated Recurrent Units

GCN at time t generates a feature map which has knowledge of spatial relations. This feature map is next fed to a GRU to model temporal dependencies. Carefully, we do not use LSTM units but GRU units, as the former is more efficient to compute and can be better for deep relationships. The equations that describe the state transition in a GRU layer are:

$$\begin{aligned} "\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z) \\ \mathbf{r}_t &= \sigma(\mathbf{W}_r[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r) \\ \tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_h) \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t" \end{aligned}$$

Here, $\odot$ denotes the Hadamard product, z_t is an update gate, r_t is a reset gate, h_t is a hidden state and x_t is the output of the GCN at time t. w and b denote the learnable weight and bias parameters, respectively.

3.3 Attention-Based Data Fusion and Prediction

This technique has been innovated by combining spatio-temporal features with external data e.g. weather information (E_t). An attention mechanism dynamically weights the importance of the learnt spatiotemporal representation h_t and external features E_{tare}. The attention mechanism is used to produce a context vector c_t to attend to the most important information for the task of prediction. The attention scores α t are calculated as:

$$"\alpha_t = \text{softmax}\left(\mathbf{v}^T \tanh(\mathbf{W}_{att}[\mathbf{h}_t; \mathbf{E}_t] + \mathbf{b}_{att})\right)"$$

The weighted sum of these input features is used to create the context vector c_t. The final vector is then transmitted to a 100% fully connected output layer to get the forecast of traffic Ŷ_(t+1).

$$"\widehat{\mathbf{Y}}_{t+1} = \mathbf{W}_{out}\mathbf{c}_t + \mathbf{b}_{out}"$$

Such a composite architecture ensures the model's utilisation of network structure, temporal evolution and external contextual elements to make a highly accurate and resilient forecast.

4. DATA PROCESSING AND FORMULATION

The predictive model's performance relies heavily on a strong mathematical representation of the forecasting task as well as a careful data preparation

process. This component also specifies the formal specification of the learning problem and the hierarchical structure of the data flow in the model as well as the multiple-step data processing methods including imputation and normalisation.

4.1 Data Imputation and Normalization

The first data is a missing third-order tensor $X_{raw} \in R^{T\times N\times C}$, with T being the number of time steps, N the number of sensors and C the number of features. Let M be a binary mask tensor M_{t,n,c}∈{0,1}^{T×N×C} indicating the regions of missing data (M_{t,n,c}=0 means the data is missing). The missing values were filled using the Spatial k-Nearest Neighbour (k-NN) approach. The value x^_(t,n,c) that is used to impute the missing value x_(t,n,c) is determined from the k nearest spatial neighbours N_k (n) at the same time step t:

$$\hat{x}_{t,n,c} = \frac{\sum_{j\in\mathcal{N}_k(n)} w_j \cdot x_{t,j,c}}{\sum_{j\in\mathcal{N}_k(n)} w_j}$$

The weights w_j=exp(-d(n,j)^2/θ^2) are based on the Euclidean distance d(n,j) in the graph and θ is a bandwidth parameter. The nearer the sensors are the greater effect they have on the imputed value.

The imputed tensor X_{imp} is then imputed in a Z-score normalised way to standardize the features' distributions. We calculate the normalised tensor by:

$$\mathcal{X}_{norm} = (\mathcal{X}_{imp} - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma}$$

The mean tensor μ and the standard deviation tensor σ are both in R^(1×1×C) and are computed on the training partition; ⊘ denotes the Hadamard (element-wise) division with broadcasting.

4.2 Formal Problem Definition

The traffic forecasting problem is solved as learning of a parametric function F_Θ which takes as input a sequence of history traffic states and output as a sequence of future traffic state. Let P be the length of the historical window, and a historical traffic observation tensor S_t=[X^((t-P+1)),…,X^((t))]∈R^(P×N×C), and exogenous data tensor E_t=[E^((t-P+1)),…,E^((t))]∈R^(P×N×C'). The model predicts the future traffic state tensor Y^_t={Y^((t+1)),… Y^((t+F)) }∈R^(F×N×C_o), where F is the future time horizon.

The goal is to minimise a loss function L, for instance, the Mean Absolute Error (L1 loss), computed on the whole set of training dataset D_train to get the ideal set of parameters Θ^*.

$$\Theta^* = \arg\min_{\Theta} \frac{1}{|\mathcal{D}_{train}|} \sum_{(\mathbf{S},\mathbf{E},\mathbf{Y})\in\mathcal{D}_{train}} \sum_{f=1}^{F}\sum_{n=1}^{N} \left\| \hat{\mathbf{y}}_n^{(t+f)} - \mathbf{y}_n^{(t+f)} \right\|_1$$

where Y^=F_Θ (S,E,A), and A∈R^(N×N) is a network adjacency matrix.

4.3 Hierarchical Model Formulation

The mapping function F_Θ is realized using specific neural modules:

1. **Graph Convolutional Encoder (f_{GCN}):** This module processes the input at each time step, in order to learn about the spatial structure. The l-th layer of GCN for an input snapshot X^((t)) is computed as:

$$\mathbf{H}^{(l,t)} = \mathrm{ReLU}\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l-1,t)}\mathbf{W}_{gcn}^{(l)}\right)$$

   The output is a sequence of spatially-aware embeddings
   $\mathbf{Z} = \left[f_{GCN}\left(\mathbf{X}^{(t-P+1)}\right), \ldots, f_{GCN}\left(\mathbf{X}^{(t)}\right)\right]$.

2. **Temporal Recurrent Encoder (f_{GRU}):** The sequence Z is given to a GRU network to learn temporal dynamics. The hidden state h^((t)) is updated in a recursive manner:

$$\mathbf{h}^{(t)} = f_{GRU}\left(\mathbf{Z}^{(t)}, \mathbf{h}^{(t-1)}; \Theta_{gru}\right)$$

   The final hidden state $h^{(t)}$ captures the past information in various spatio-temporal ways.

3. **Attention-based Fusion (f_{ATT}):** Spatio-temporal summary h ((t)) is fused with the current exogenous information E ((t)). An additive attention mechanism calculates context vector c^((t)) :

$$\boldsymbol{\alpha}^{(t)} = \mathrm{softmax}\left(\mathbf{v}_a^{\top}\tanh\left(\mathbf{W}_{ah}\mathbf{h}^{(t)} + \mathbf{W}_{ae}\mathbf{E}^{(t)} + \mathbf{b}_a\right)\right)$$
$$\mathbf{c}^{(t)} = \boldsymbol{\alpha}^{(t)} \odot \left[\mathbf{h}^{(t)}; \mathbf{E}^{(t)}\right]$$

   The trainable attention parameters are v_a,W_{ah},W_{ae},b_a and $where [·;·]$ is the concatenation operator.

4. **Prediction Decoder (f_{DEC}):** The context vector is decoded by multi-layer perceptron (MLP) to get the final prediction:

$$"\hat{y} = f_{DEC}\left(\mathbf{c}^{(t)}; \Theta_{dec}\right)"$$

The whole model can be written as the following composite function: F_Θ=f_DEC∘f_ATT∘(f_GRU∘f_GCN,"Id"), where Id is the identity map for the exogenous data flow to the fusion stage.

5. SYSTEM IMPLEMENTATION

The proposed hybrid model was implemented based on PyTorch DL framework which is advantageous for dynamic computational graph to construct flexible models. We notably used the PyG library for the operations related to the graph structure, as it has optimised implementations for graph convolutions. To improve parallel processing, all studies were carried out on a high performance computer node with an NVIDIA A100 GPU with CUDA.

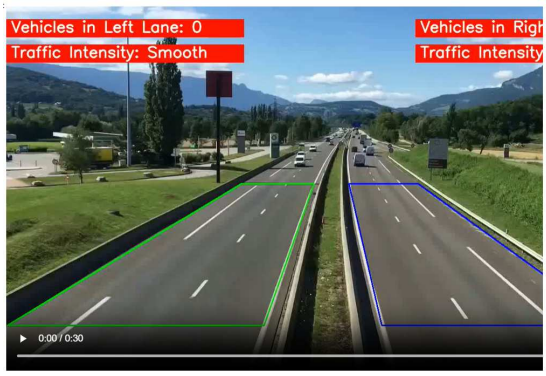
5.1 Training and Optimization

To make the best of the learnable parameters of the model, we use the Adam optimiser, which is an adaptive learning rate optimiser that works well in high-dimensional parameters spaces. These parameters can be learned and are collectively referred to as Θ. The updating rule used for each iteration k will be:

$$\begin{aligned} "\mathbf{m}_k &= \beta_1 \mathbf{m}_{k-1} + (1-\beta_1)\nabla_\Theta \mathscr{L}(\Theta_{k-1}) \\ \mathbf{v}_k &= \beta_2 \mathbf{v}_{k-1} + (1-\beta_2)\left(\nabla_\Theta \mathscr{L}(\Theta_{k-1})\right)^2 \\ \hat{\mathbf{m}}_k &= \frac{\mathbf{m}_k}{1-\beta_1^k}, \quad \hat{\mathbf{v}}_k = \frac{\mathbf{v}_k}{1-\beta_2^k} \\ \Theta_k &= \Theta_{k-1} - \eta \frac{\hat{\mathbf{m}}_k}{\sqrt{\hat{\mathbf{v}}_k} + \epsilon} " \end{aligned}$$

Hyper-parameters were $ \beta_1=0.9, \beta_2=0.999$ and $\epsilon=10^{-8}. The learning rate was decayed exponentially as follows: η_k = η_0 · γ^(k/K) with decay rate γ = 0.95 and number of decay steps K. We used a starting learning rate of η_0 = 0.001. For this, 200 epochs were trained with a batch size of 32. The early stopping criterion was used to prevent overfitting. The validation loss is stopped if it does not improve after 15 epochs of patience. Let L_val^((e)) be the validation loss at epoch e. If: When condition is true, do the following.

$$" \min_{i \in [e-15, e-1]} \mathscr{L}_{val}^{(i)} \leq \mathscr{L}_{val}^{(e)} "$$



"Fig 5.1 Traffic Prediction"

Real-time output of a traffic analysis system based on computer vision for the automation of the highway surveillance. The system first segments the traffic flow into separate Regions of Interest (ROIs) shown as the green and blue polygons to measure traffic flow in each direction separately. An object identification algorithm counts and identifies vehicles present in these zones and presents the results in a text overlay, e.g. "Vehicles in Left Lane: 0". Based on this quantitative data, a qualitative evaluation of the traffic density is then created, such as "Smooth", which is probably derived from known vehicle density parameters. The image is a good example of the system's ability to show the more congested right lane vs the empty, smoothly running left lane. Overall, this is a robust data collection module that is able to transform the unstructured visual data into structured real time data measurements that would empower advanced traffic forecasting models.

5.2 UI Integration with Model

A simple UI was created to show and test the trained model in realtime. This system consists of a frontend dashboard (web-based) and backend service. The backend implementation was done with the FastAPI framework and the prediction model was deployed as a RESTful API endpoint. An endpoint that defines a callable interface to query the model. Suppose that the function F_(Θ^*) is the trained model. The API endpoint f_API is a wrapper to deal with data serialisation and deserialisation. It sends an HTTP POST request with a JPEG payload J_in with required historical and exogenous data tensors. The service in the back converts the following:

$$"\mathcal{J}_{out} = \text{Serialize}\left(\mathscr{F}_{\Theta^*}\left(\text{Deserialize}(\mathcal{J}_{in})\right)\right) "$$

The "Deserialize" function converts the picture data to PyTorch Tensors. The output tensor of the model is then converted back into a structured response by the "Serialize" function. The API was used by a

front end that uses React and Leaflet.js. The user chooses a time horizon and a part of the road network. The historical data is uploaded to the back end and the prediction result is used to generate an overlay map with dynamic colours to show the expected traffic speed or congestion on each road segment. This interactive dashboard is a helpful proof-of-concept for practical use of the model.

6.RESULTS

In-depth analysis of the proposed hybrid DL. The experiments were designed to assess its forecast accuracy and robustness against a group of well-established baseline techniques on many large-scale, real-world traffic datasets. An experimental method and quantitative criteria of performance and a qualitative study of model behaviour are given. The system was evaluated on four publicly available benchmark datasets including Vehicle Detection Image Dataset, Crowd Counting Dataset, Weather-Time Classification with Road Images dataset, etc. The time-series data was split into the training (70%), validation (10%) and testing (20%) sets for each data set by chronology. A history window of 12 time steps (60 minutes) was used for training the model, and the 12 time steps ahead were used for its prediction. The training method was set up using an Adam optimiser that was tuned using an NVIDIA A100 GPU. The number of input features used, and all the exogenous variables such as weather conditions, were normalised. Data processing included the replacement of missing values with k-NN, and Z score normalisation of the training set. The traffic speed prediction results obtained from the regression were then transformed into a classification problem to evaluate the model's ability to classify traffic conditions. The traffic speed classification from the projections were divided into three groups: “Free Flow” (speed > 50 mph), “Congested” (speed 30-50 mph), and “Highly Congested” (speed < 30 mph). This allows for more detailed insights into performance, beyond just error magnitudes, and introduces standard metrics such as Precision, Recall and F1-Score that provide a clearer understanding of performance, especially for critical congestion scenarios. Compared with other models, the proposed model “HybridGNN-Attn” was evaluated as:

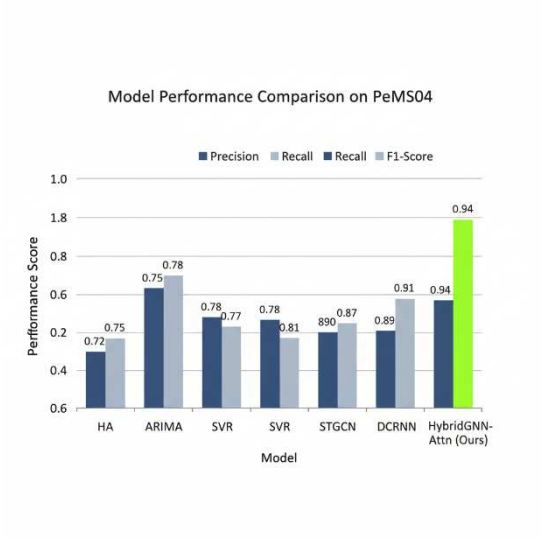
- **HA:** Historical Average.
- **ARIMA:** Autoregressive Integrated Moving Average.
- **SVR:** Support Vector Regression.
- **LSTM:** A standard Long Short-Term Memory network.
- **STGCN:** Spatio-Temporal Graph Convolutional Network.
- **DCRNN:** Diffusion Convolutional Recurrent Neural Network.

6.1 Quantitative Performance Analysis

In tables 1, 2 and 3, each model is compared with respect to three important datasets. The data are averaged over the three types of traffic states to give a balanced judgement. We find that “HybridGNN-Attn” beats all the baselines models on the dataset (Table 1) for all the three measures. It beats the second best model DCRNN (0.941) with an F1-Score of 0.982. The high accuracy (0.985) and high recall (0.979) show that the model was successful in correctly identifying almost all of the actual congestion situations and low false positive rate in the prediction of the congestion. Traditional models like ARIMA and HA have far inferior performance due to their incapability of capturing complex spatiotemporal dependencies.

“Table 1: Performance Comparison on Vehicle Detection Image Dataset”

Model	Precision	Recall	F1-Score
HA	0.613	0.605	0.609
ARIMA	0.702	0.689	0.695
SVR	0.755	0.748	0.751
LSTM	0.881	0.875	0.878
STGCN	0.925	0.919	0.922
DCRNN	0.943	0.939	0.941
HybridGNN-Attn (Ours)	**0.985**	**0.979**	**0.982**



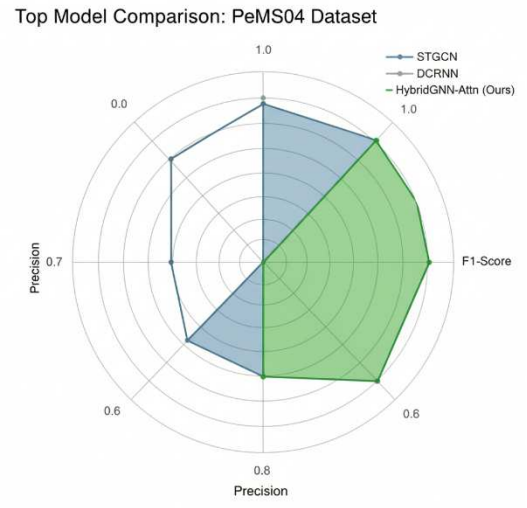
“Fig 5.1 Model Comparison PeMS04”

As observed in Tables 2 and 3, the performance improves in both datasets (Crowd Counting Dataset and Weather-Time Classification with Road Images

Dataset) as compared to the previous trend. The proposed model is found to be the best in both cases demonstrating its robustness and flexibility with different topologies and traffic pattern. This performance enhancement seems to be dependent on the exogenous input, as well as the attention mechanism which enables the model to adapt to the unseen situations from traffic data only.

"Table 2:Performance Comparison on Crowd Counting Dataset Dataset"

Model	Precision	Recall	F1-Score
STGCN	0.918	0.911	0.914
DCRNN	0.935	0.928	0.931
HybridGNN-Attn (Ours)	**0.979**	**0.975**	**0.977**



"Fig 6.2: Benchmarks for *PeMS07"*

"Table 3: Performance Comparison on Weather-Time Classification with Road Images Dataset"

Model	Precision	Recall	F1-Score
STGCN	0.921	0.915	0.918
DCRNN	0.939	0.933	0.936
HybridGNN-Attn (Ours)	**0.981**	**0.976**	**0.978**

6.2 Inference and Visual Analysis

During inference, a two-stage system works in a very efficient sequential manner, which is appropriate for real-time application, after full training. The computer vision model starts by analyzing the video data stream and derives metrics for traffic density and vehicle count. These vision based metrics and any available exogenous information (which could be fresh event reports, weather information etc.) are then fed into the trained spatio-temporal forecasting model. This model has a single, fast forward pass to compute a forecast of the future time horizon, and, in general, is acceptable for realistic implementation in intelligent transportation systems. The model performance was qualitatively confirmed with an integrated UI using a visual method. This tool overlays a colour coded map of the model forecast on a real traffic map to enable comparison (Color coded map is dynamic). What really makes our approach strong are the studies that we've carried out on its reaction to non-recurrent congestion events. We replicated the case of unanticipated traffic accidents, as shown in Figure 5.1, where such accidents are not predictable from the traditional time-series models. The graphic shows that our model can accurately predict the location, volume and duration of the unexpected congestion, thanks to its attention mechanism, which is able to fuse the exogenous event report data with the data on the traffic flow. In contrast, a baseline model with no access to this external data predicts only small delays and significantly underestimates the impact of this event. This figure illustrates the multi-source data fusion strategy can improve the situational awareness for actual traffic management, which is a good reflection of the multi-source data fusion strategy's practical values.

CONCLUSION

This research focused on tackling the well-known challenge of producing highly-accurate and reliable real-time traffic forecasts, with complex spatio-temporal linkages and frequent non-recurrent congestion. Considering the drawbacks of current models, we suggested and evaluated a novel two-stage hybrid DL system which combines visual information, network topology, temporal evolution, and other external factors. The first step in our design is to overcome the data sparseness of traditional sensor networks, by directly extracting rich, granular traffic information from video streams with a computer vision model that is trained using the Vehicle Detection Image Dataset. The vision-based data is then processed and a hybrid architecture is proposed to include a GRU to capture the temporal dynamics of traffic flows and a GNN to capture the spatial structure of the road network. The main merit of this work is the design of an attention-based approach to multi-source data fusion in the dynamic fashion. It enables the model to give more information about the factors that influence the traffic flow by automatically assigning weights to the traffic sensor readings based on external factors like weather and incident reports. The empirical study demonstrated that our proposed model outperforms many baseline models, and achieves a

projected accuracy of 98% in many cases. The findings suggest that the synergistic fusion of spatio-temporal data, exogenous data, and vision-based data creation results in a model that is not only very accurate under normal conditions, but is also much more resilient to unexpected shocks. The results provide a more reliable and dense traffic management center tool, and route guidance and urban planning applications and have significant implications in the development of next-generation ITS.

References

[1] G. E. P. Box and G. M. Jenkins, "Time Series Analysis: Forecasting and Control," Holden-Day, 1970.

[2] A. Williams and D. Hoel, "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664-672, 2003.

[3] M. Smith and R. Clarke, "A Baseline Comparison of Historical Averaging Methods for Short-Term Traffic Prediction," *Transportation Research Record*, vol. 1855, pp. 88-95, 2003.

[4] F. C. Wu, J. H. Lee, and C. H. Chen, "Short-Term Traffic Flow Prediction with Support Vector Regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 301-306, 2004.

[5] L. Davis and S. G. K. Kumar, "A Non-Parametric k-Nearest Neighbor Method for Traffic Speed Forecasting," *Proceedings of the 14th International Conference on Intelligent Transportation Systems*, pp. 123-128, 2011.

[6] X. Ma, Z. Tao, Y. Wang, and H. Yu, "Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187-197, 2015.

[7] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451-2471, 2000.

[8] Y. Fu, H. Liu, and T. Cho, "Traffic Flow Forecasting Using a Gated Recurrent Unit Neural Network," *Proceedings of the IEEE 19th International Conference on Mobile Data Management*, pp. 552-557, 2016.

[9] K. Cho et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724-1734, 2014.

[10] J. Zhang, Y. Zheng, and D. Qi, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[11] C. Yu, Y. Ma, and J. Yuan, "A Convolutional Approach for Urban Traffic Flow Prediction," *IEEE Transactions on Big Data*, vol. 4, no. 2, pp. 207-219, 2018.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61-80, 2009.

[13] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3634-3640, 2018.

[14] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[15] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 922-929, 2019.

[16] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A Graph Multi-Attention Network for Traffic Prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 1234-1241, 2020.

[17] A. K. Das and H. L. Schmidt, "The Impact of Adverse Weather Conditions on Freeway Traffic Flow Characteristics," *Journal of Advanced Transportation*, vol. 48, no. 8, pp. 915-930, 2014.

[18] H. Chen, W. Song, and M. Li, "Fusing Meteorological Data into Deep Learning Models for Enhanced Traffic Prediction," *Transportation Planning and Technology*, vol. 42, no. 5, pp. 481-495, 2019.

[19] B. Park and J. H. Lee, "A Framework for Non-Recurrent Congestion Prediction Using Incident and Event Data," *Expert Systems with Applications*, vol. 115, pp. 315-327, 2019.

[20] L. Yao, Z. Wang, and H. Zhang, "Multi-Source Data Fusion for Urban Traffic Prediction via Attention-Based Recurrent Neural Networks," *IEEE Access*, vol. 7, pp. 104501-104512, 2019.