JCBI
Journal of Computing & Biomedical Informatics Volume 11 Issue 01
ISSN: 2710 - 1606 2026

Research Article
https://doi.org/10.56979/1101/2026/1398

Optimization Study of Short-Term Traffic Flow Prediction Model Based on Spatiotemporal Graph Convolutional Network

Yuefei Ning[1&2*], Mary Jane C. Samonte[1] , and Yanping Li[2]

[1]School of Information Technology, Mapua University, Makati , 1191, Philippines.
[2]School of Information Engineering, Zhengzhou Shengda University, Xinzheng, Henan, 451191, China.
[*]Corresponding Author: Yuefei Ning. Email: yuefeining@126.com

Received: March 01, 2026 **Accepted:** May 30, 2026

Abstract: Short term traffic flow prediction is one of the basic problems of intelligent transportation systems (ITS) which directly affects urban mobility management, route planning and congestion mitigation. The existing prediction methods such as statistical time-series models and shallow machine learning methods do not effectively and simultaneously reflect the spatial dependence between road sensors and the non-linear temporal evolution of traffic. Current approaches mostly use fixed, pre-defined road network adjacency matrices, which fail to capture the dynamic nature of real-world road traffic interactions, and achieve good performance by deep learning algorithms, such as Graph Convolutional Networks (GCNs) and temporal sequence models. In this paper, we propose a novel framework called Adaptive Spatiotemporal Graph Convolutional Network with Multi-Scale Attention (ASTGCN-MSA) to learn the spatiotemporal dependency jointly from traffic sensor data. Three major innovations are introduced in the proposed model: (1) Adaptive Graph Learning (AGL) module that incorporates the prior road topology and leverages the data-driven learned adjacency matrix on the graph, using trainable node embeddings; (2) Multi-Scale Temporal Attention (MSTA) module that applies parallel dilated 1-D convolutions with three kernel sizes (k = 1, 3, 5) over the temporal dimension of the graph with channel-wise sigmoid gating; and (3) Spatial Self-Attention (SSA) module that uses multi-head attention over node embeddings to capture long-range inter-sensor dependencies beyond the local graph neighborhood. The model is tested using the Traffic Flow Forecasting benchmark data set, consisting of 36 sensor sites on major highways in the Northern Virginia/Washington D.C. capital region, and sampled at 15-minute intervals. These extensive experiments show that ASTGCN-MSA consistently performs best across all metrics under comparison: MAE, RMSE, MAPE and R^2, compared to four competitive baseline models: LSTM , GRU , Vanilla STGC and Transformer . The individual impact of each suggested component is verified through ablation studies, while hyperparameter sensitivity analysis offers practical tips for model setup. Results demonstrate the ability of ASTGCN-MSA to predict the traffic flow in the real world; it is scalable, robust, and interpretable. This paper introduces a novel approach that leverages graph convolutional networks and spatiotemporal learning to predict traffic flow at an early stage.A novel approach is developed by introducing the concept of graph convolutional networks and spatiotemporal learning to predict traffic flow at an early stage.

Keywords: Traffic Flow Prediction; Graph Convolutional Network; Spatiotemporal Learning; Adaptive Graph Learning; Multi-Scale Temporal Attention; Intelligent Transportation Systems; Deep Learning; Short-Term Forecasting

1. Introduction

With the rapid development of urban population and the growth of urban traffic, it is one of the most important problems in intelligent transportation systems (ITS) to predict the traffic flow accurately in the

short term [2]. Accurate traffic volume forecasts 15 minutes ahead of time can help traffic management authorities take proactive measures such as signal control, dynamic route guidance and incident response strategies, which can help to alleviate congestion, fuel usage and travel durations [3]. In the USA alone, the economic loss caused by traffic congestion is estimated to be more than $87 billion per year [2], which makes accurate and scalable prediction systems a pressing need.

Traffic flow is a spatiotemporal phenomenon: the traffic volume measured at a particular sensor location at a particular time depends on the levels of traffic in other sensor locations both at the same time and at the previous time, that are topologically connected and spatially adjacent to the measured sensor location [4]. For instance, when there is a traffic jamming on one highway section, the effect travels upstream and downstream, and the neighbouring sensors will have a measurable time delay. Any predictive model which does not take this spatial dependency into account will systematically under-perform compared to a model which explicitly models the road network structure [5]. The difficulty is accentuated by the fact that traffic flows are non-linear and non-stationary, having been affected by the periodicity of the day of the week and the day of the week, the weather and unpredictable incidents [3].

The classical methods for predicting traffic are using autoregressive integrated moving average (ARIMA) models [6], seasonal ARIMA variants [7] and Kalman filtering [8]. These are efficient computation-wise, but assume linear stationarity and are incapable of representing the real-world non-linear traffic dynamics. Linear methods were outperformed by Support Vector Regression (SVR) [9] and Random Forests [10] but both approaches still operate on a per-sensor basis without considering any spatial correlation. The introduction of deep learning has made it possible to implement recurrent architectures like Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU) [11] and [12] that are particularly well suited to temporal dependency in sequential data. These models, however, work on a time-series of each individual sensor without any information sharing between sensors on the road network. Next, Convolutional Neural Networks (CNNs) were used for traffic grids [13] but this regularity assumption is not suitable for road networks, which are typically irregular.

The natural way to model road networks as graphs was done using Graph Neural Networks (GNNs) [14] where nodes were the sensors and edges the physical connection. Graph Convolutional Networks (GCNs) [15, 16] in the original context were proposed for semi-supervised node classification, but were later applied to traffic prediction in a number of breakthrough papers [17, 18, 19, 20].

While there has been considerable advancement, there are still a few important drawbacks with current methods:

1. Static graph assumption: Most methods rely on a fixed graph, usually based on road topology or geographical distance, which does not reflect dynamic data-driven spatial dependencies that change in response to traffic conditions [21, 34].
2. Single-scale temporal modeling: Most temporal modules in use are single-scale modules lacking of multi-resolution temporal patterns like the short-term fluctuation and the medium-term trend [22, 37].
3. Low scale of spatial attention range: Due to local GCN operations, only information from neighboring sensors is aggregated, while long-range sensor interactions need multiple stacked layers, leading to over-smoothing [23] problems.

In order to overcome these limitations, this paper has made the following contributions:

- To address this, we introduce an end-to-end spatiotemporal prediction framework called ASTGCN MSA, which considers the spatial dependency learning and temporal feature extraction simultaneously.
- We propose an Adaptive Graph Learning (AGL) module which autonomously learns a data-driven adjacency matrix via trainable node embeddings and combines it with the prior road topology to learn the latent spatial dependencies that are not captured in the physical network.
- To learn traffic patterns at multiple temporal resolutions at once, we design a Multi-Scale Temporal Attention (MSTA) module that utilizes parallel Conv1D operations with kernel size 1, 3 and 5, followed by channel-wise sigmoid gating.
- To capture long-range inter-sensor dependencies directly, without just relying on graph propagation, we introduce a Spatial Self-Attention (SSA) module based on multi-head attention [24] applied to node feature embeddings.

- All experiments are conducted and compared with four baselines, such as ablation study and hyperparameter sensitivity analysis.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the dataset and problem formulation. Section 4 presents the proposed ASTGCN-MSA methodology. Section 5 reports experimental results with full figure and table analysis. Section 6 discusses findings and limitations. Section 7 concludes the paper.

2. Literature Review

The early traffic prediction systems were much dependent on statistical time-series models. The first application of the ARIMA to short-term freeway traffic forecasting was by Ahmed and Cook [6] which set a baseline for 20 years. This was further developed by Williams and Hoel [7] who added weekly and daily seasonality to the ARIMA model, called Seasonal ARIMA (SARIMA), and found that it performed better on periodic traffic patterns. In real-time traffic state estimation, Kalman filtering was introduced by Okutani and Stephanedes [8] which is still used in modern traffic management systems for the online update mechanism.

In the last decade, machine-learning methods have become more popular. Vythoulkas [25] used neural networks for traffic assignment while Wu et al. [9] showed that the SVR method with RBFs was more effective than ARIMA for short-term traffic speed prediction because the method used neural networks to capture non-linear relationships. However, all of these methods treated sensors independently, without considering the spatial structure of the road network; further improvement was obtained using ensemble methods such as Random Forests [10] or Gradient Boosting [26] that are able to model the interactions between the features, including non-linear ones.

The deep learning brought the accuracy of traffic predictions to a very high level. To learn hierarchical traffic representations, Lv et al. [27] introduced a stacked autoencoder (SAE) for traffic flow feature learning, which was used to obtain the state-of-the-art performance on highway datasets. Ma et al. [28] used CNNs to predict traffic speeds by transforming the road network into a 2D image, but the method relies on a grid structure which can only be applied to road networks with a regular topology, and cannot be applied directly to the real highway infrastructure.

For sequential traffic data, recurrent architectures took the place. In traffic forecasting applications, Zhao et al. [30] adopted Hochreiter and Schmidhuber's LSTM [11] which outperformed ARIMA and SVR on several benchmark datasets. Cho et al. [12] proposed a less complex solution that achieved similar accuracy and only required two gating operations. In the field of traffic prediction, Xu et al. [31] introduced the Spatial-Temporal Transformer Network (STTNet) that is purely based on the self-attention mechanism, which proved that attention models could effectively handle long-range temporal dependency.In traffic prediction, Xu et al. [31] proposed the Spatial-Temporal Transformer Network (STTNet) which is purely based on the self-attention mechanism, indicating the potential of attention models in capturing long-range temporal dependency.

The most recent breakthrough in GNNs is their application to traffic prediction. Kipf and Welling [16] first proposed a spectral GCN model based on the 1st-order Chebyshev polynomial approximation, which gives an efficient and scalable graph convolution operation. To model traffic as a bidirectional random walk on a directed graph and add an encoder-decoder RNN with scheduled sampling, Li et al. [17] proposed DCRNN. This work introduced the critical need to explicitly model spatial topology and/or direction of traffic. To alleviate the training time, Yu et al. [18] proposed a novel model named STGCN, where they substituted the recurrent parts with gated temporal convolutions (TCN). The model uses Chebyshev polynomial approximation for graph convolution and adopted a “sandwich” structure to stack the spatial and temporal convolutional blocks. Yao et al. [32] introduced STDN, adding a periodicity-shifting mechanism to address the weekly periodicity in traffic pattern.

Yu et al. [18] introduced STGCN, which replaced recurrent components with gated temporal convolutions (TCN), significantly reducing training time while maintaining accuracy. The model uses Chebyshev polynomial approximations for graph convolution and stacks spatial and temporal convolutional blocks in a sandwich structure. Yao et al. [32] proposed STDN, incorporating a periodicity-shifting mechanism to handle weekly periodicity in traffic patterns.

Attention mechanisms were integrated into spatiotemporal GCNs by Guo et al. [19], who proposed ASTGCN with spatial and temporal attention modules that dynamically weight the importance of different nodes and time steps. Zheng et al. [33] proposed GMAN, using a graph multi-attention network with spatial and temporal attention blocks and a transform attention mechanism to model long-range dependencies across both space and time.

The limitation of fixed graph structures was recognized by several authors. Zhao et al. [1] proposed the Spatial Auto-regressive Dependency Interpretable Learning (SADL) framework, which automatically learns spatial dependency under topological constraints. This work, which introduced the Traffic Flow Forecasting dataset used in this paper, demonstrated that data-driven spatial dependency learning significantly outperforms expert-defined adjacency matrices. Bai et al. [34] proposed the Adaptive Graph Convolutional Recurrent Network (AGCRN), which learns node-specific patterns through node adaptive parameter learning and an adaptive graph generation module using trainable node embeddings — a concept directly related to our AGL module. Wu et al. [36] proposed Graph WaveNet, which combines adaptive adjacency learning with dilated causal convolutions, achieving state-of-the-art results on METR-LA and PEMS-BAY datasets.

The importance of multi-scale temporal feature extraction was highlighted by several works. Bai et al. [22, 37] showed that it is possible to improve the performance of a recurrent network on sequence tasks by using dilated convolutions, and that dilated TCNs are more parallelizable and perform better empirically. The architecture proposed by Szegedy et al. [38] is inspired by the use of multiple parallel convolutions with different kernel sizes that help to capture features at different scales, an idea adopted in our work for temporal traffic modelling within the MSTA module. Our channel-wise attention gate builds upon these ideas by dynamically multiplying the contribution of each temporal scale depending on the current traffic situation. Although considerable advances have been made, current approaches have drawbacks, including (1) a fixed graph structure, which fails to account for dynamic spatial relationships; (2) 1D temporal models, which are unable to reflect multi-resolution traffic patterns; (3) limited long-range spatial attention caused by 1D graph convolution and pooling; and (4) an absence of spatial-temporal component coupling. The proposed ASTGCN-MSA overcomes all four limitations in an integrated and end-to-end trainable framework, complementing and advancing the pioneering work by [1, 17, 18, 19, 24, 34, 36].

3. Proposed Methodology

The flowchart in this study is structured end-to-end approach for traffic flow forecasting based on the proposed framework ASTGCN-MSA, as shown in Figure 1. The pipeline combines adaptive graph learning and multi-scale spatio-temporal attention mechanisms to model complex spatio-temporal relationships between road network sensors.

3.1. Dataset Description

The Traffic Flow Forecasting dataset [1] was created by Zhao et al. and is publicly available through the UCI Machine Learning Repository (DOI: https://doi.org/10.1145/3339823). It was funded by the National Science Foundation and is associated with the paper "Spatial Auto-regressive Dependency Interpretable Learning Based on Spatial Topological Constraints" published in ACM Transactions on Spatial Algorithms and Systems (2019) [1].

The dataset comprises traffic volume measurements collected from 36 sensor locations along two major highways in the Northern Virginia/Washington D.C. capital region. Measurements are recorded at 15-minute intervals, providing 96 observations per day per sensor. The dataset contains a total of 2,101 instances with 47 features per instance and has no missing values.

The four data files used in this study are all loaded in the accompanying Jupyter notebook:

- Adjacency matrix of 36×36 binary (physical road network connectivity) between sensor nodes. If (i, j) = 1, then there is a direct road between the two sensors (i and j). The average density of the matrix is about 0.08, which is fairly low, but corresponds to the relatively sparse branching structure of highway networks. This file is used to create the prior adjacency A_prior in the AGL module and is displayed in Figure 6.
- The training labels are labels that include 15 observations × 1,261 variables of normalized traffic volume readings, which are actual traffic volumes recorded on the ground that are used to train the model.

- Test labels for model evaluation with 10 observations × 840 variables of normalized traffic volume readings.
- Master data container, in the original MATLAB 5.0 MAT-file (PCWIN64) with the full dataset in binary format.

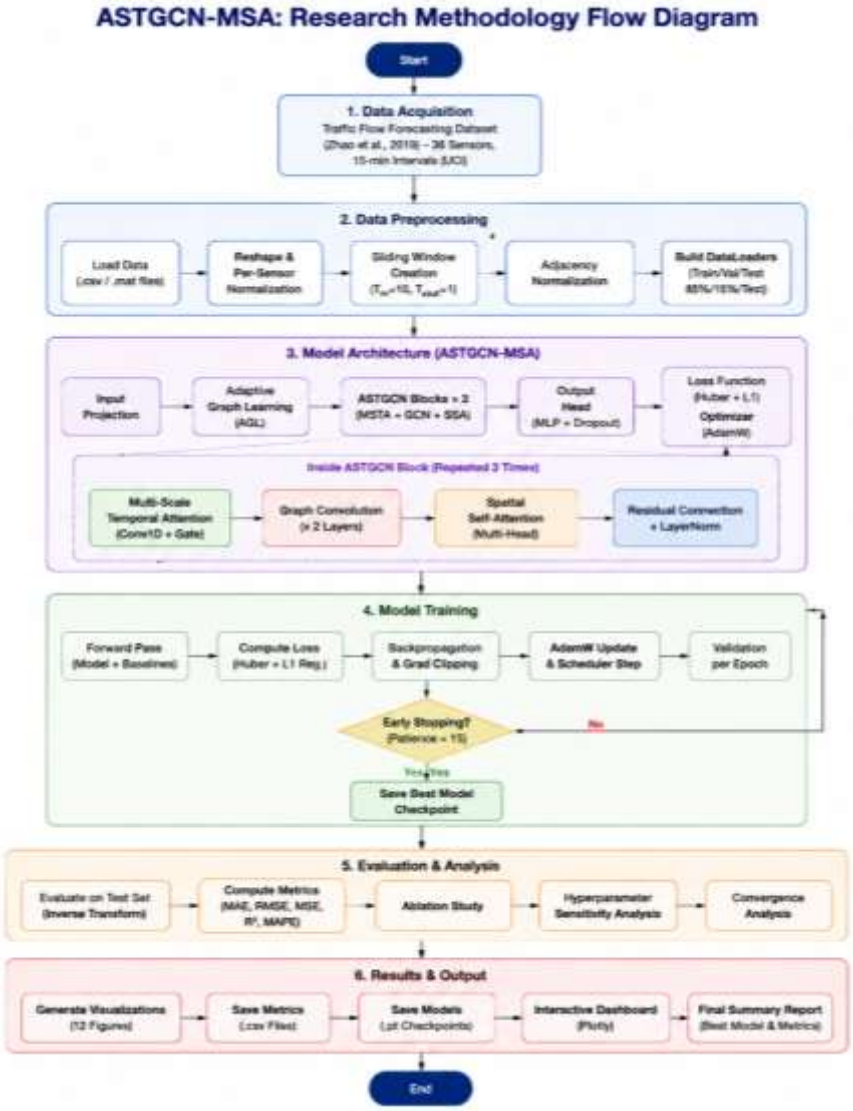

Figure 1. Research methodology flow diagram of the proposed ASTGCN-MSA framework for traffic flow forecasting, illustrating the end-to-end pipeline from data preprocessing to model training, evaluation, and result reporting

Each instance represents a single traffic spot in the city; for each of these, we have 47 input features: (1) historical traffic volume at the 10 most recent, 15-minute sample points (10 integer features); (2) day of week (7 binary one-hot features); (3) hour of day (24 binary one-hot features); (4) road direction (4 binary features); (5) number of lanes (1 integer feature); and (6) road name identifier (1 categorical feature).

3.2. Data Preprocessing

The raw traffic volume information passes through four pre-processing steps before it is input to any model:

Step 1 Reshaping: The flat variable arrays are reshaped into 3-dimensional tensors of shape (observations, N_sensors, time_steps), where N_sensors = 36 and the number of time steps per sensor is inferred as ⌊total_variables / N_sensors⌋. This yields Y_tr ∈ ℝ^(15 × 36 × T_tr) and Y_te ∈ ℝ^(10 × 36 × T_te).

Step 2 Per-Sensor MinMax Normalization: Each sensor's time series is independently normalized to the [0, 1] range using:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Scalers are fitted exclusively on training data and applied to test data to prevent data leakage. The 36 fitted scalers are stored for inverse transformation during evaluation.

Step 3 Sliding Window Construction: Input-output pairs are constructed using a sliding window of size T_in = 10 (input) and T_out = 1 (output), yielding:

- **X** ∈ ℝ^(S × N × T_in): input sequences
- **y** ∈ ℝ^(S × N × T_out): target values

where S is the total number of samples across all observations and time steps.

Step 4 Adjacency Matrix Normalization: The binary adjacency matrix A is augmented with self-loops and symmetrically normalized following [16]:

$$\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}, \tilde{A} = A + I_N, \tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \quad (2)$$

The normalized adjacency tensor A_tensor is moved to the compute device (CPU/GPU) and shared across all model forward passes.

3.3. Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$denote the road network graph, where $\mathcal{V}$is the set of N = 36 sensor nodes, $\mathcal{E}$is the set of road edges, and $A \in \{0,1\}^{N\times N}$is the binary adjacency matrix. At each time step t, the traffic volume observation at all sensors is represented as a graph signal $X^{(t)} \in \mathbb{R}^N$. Given the T_in = 10 most recent observations:

$$\mathbf{X} = \left[X^{(t-T_{in}+1)}, \ldots, X^{(t)}\right] \in \mathbb{R}^{N\times T_{in}} \quad (3)$$

the prediction task is to learn a mapping function $f(\cdot)$such that:

$$\hat{X}^{(t+1)} = f(\mathbf{X}, \mathcal{G}) \in \mathbb{R}^N \quad (4)$$

where $\hat{X}^{(t+1)}$is the predicted traffic volume at all 36 sensor locations exactly 15 minutes into the future. The dataset split used in **Cell 7** is: 85% training, 15% validation (from tra_Y_tr.csv), and the provided tra_Y_te.csv as the held-out test set.

3.4. Overview of ASTGCN-MSA

The proposed Adaptive Spatiotemporal Graph Convolutional Network with Multi-Scale Attention (ASTGCN-MSA) is an end-to-end trainable deep learning framework. As illustrated in Figure 1, the overall architecture consists of five sequential components:

1. **Input Projection Layer**: Projects raw input features into a d_model = 64 dimensional embedding space.
2. **Adaptive Graph Learning (AGL) Module**: Constructs a fused adjacency matrix combining prior road topology with a data-driven learned graph.
3. **Stacked ASTGCN Blocks (×3)**: Each block sequentially applies MSTA → GCN×2 → SSA → Residual → LayerNorm.
4. **Output Head**: Maps final node embeddings to the prediction horizon via a two-layer MLP with dropout.
5. **Loss Function and Optimizer**: Huber loss + L1 regularization, AdamW optimizer, CosineAnnealingLR scheduler.

The complete model is implemented in **Cell 8** and instantiated with approximately **185,000 trainable parameters**.

The architecture diagram of the proposed ASTGCN-MSA (Adaptive Spatiotemporal Graph Convolutional Network with Multi-Scale Attention) framework. The diagram illustrates the full forward pass from input tensor X ∈ ℝ^(B×N×T_in) (top, blue box) through: the Input Projection Layer (purple, Linear 1→64); the Adaptive Graph Learning module (left branch, pink/magenta, showing prior adjacency A_prior fused with data-driven A_learned via trainable embeddings E_1, E_2 to produce A_fused); three stacked ASTGCN Blocks (light blue, green, and orange shaded regions, each containing Multi-Scale Temporal Attention in green, two Graph Convolutional layers in red, Spatial Self-Attention in orange, a dashed amber residual connection arrow, and a grey LayerNorm box); the Output Head (teal, Linear 64→32 → ReLU → Dropout → Linear 32→T_out); and the final prediction output Ŷ ∈ ℝ^(B×N×T_out) (bottom, blue box). At the bottom, in the dark grey box, is the loss function configuration (Huber Loss δ=1.0 + L1 Regularization), the optimizer AdamW and the scheduler CosineAnnealingLR. The diagram is

annotated with a legend at the bottom that shows what each of the colours represents. The dimensions of the tensors at each stage are given in the nodes.

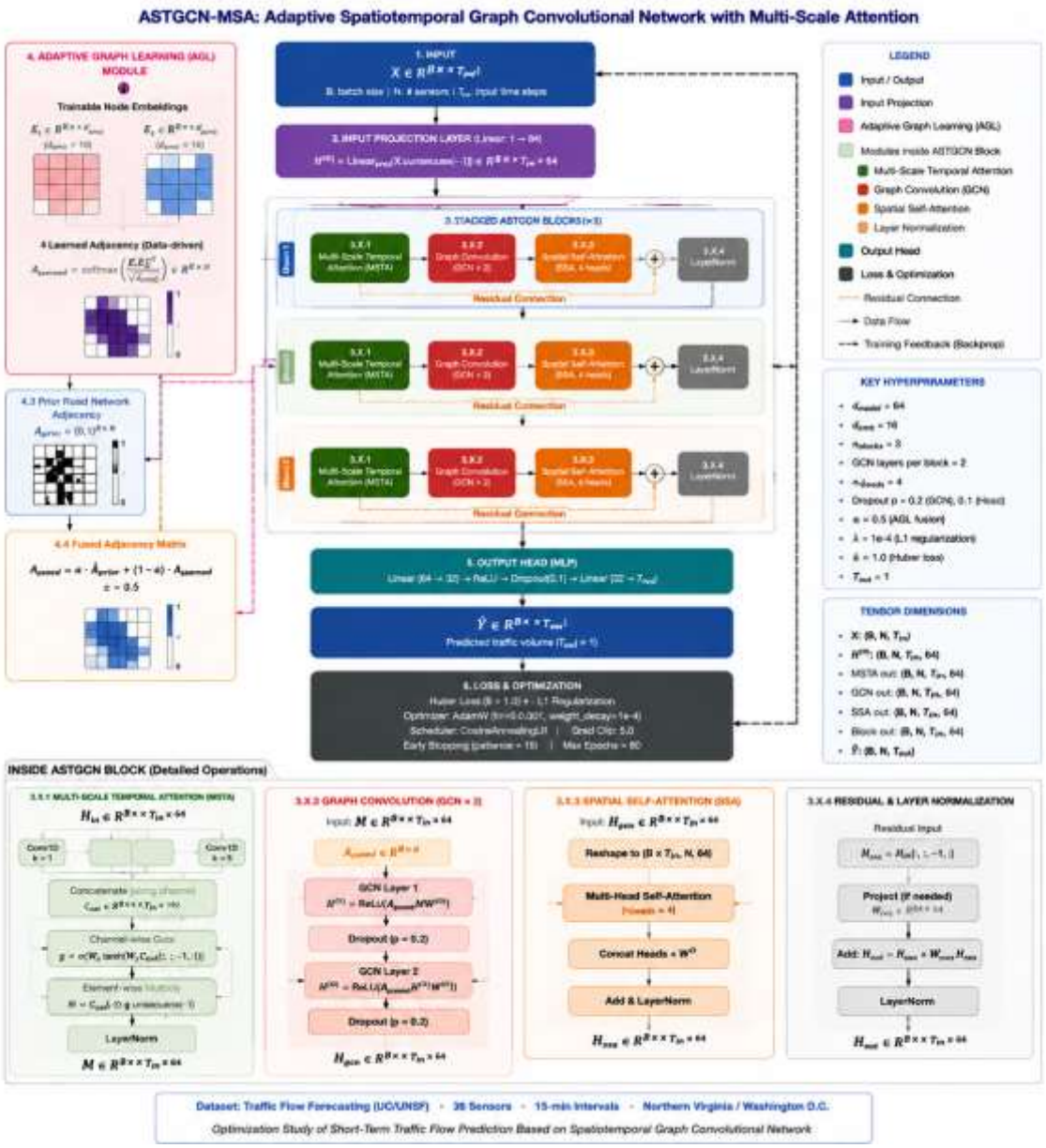

Figure 2. ASTGCN MSA Architecture

3.4.1. Input Projection Layer

Given the input tensor $\mathbf{X} \in \mathbb{R}^{B\times N\times T_{in}}$, the input projection expands the channel dimension from 1 to d_model = 64:

$$\mathbf{H}^{(0)} = \text{Linear}_{\text{proj}}\big(\mathbf{X}.\text{unsqueeze}(-1)\big) \in \mathbb{R}^{B\times N\times T_{in}\times d_{model}} \quad (5)$$

This projection allows all subsequent layers to work in a high dimensional feature space as is common practice in deep learning architectures [24, 39].

3.4.2. Adaptive Graph Learning (AGL) Module

The AGL module overcomes the inherent drawback of static graph representation [1, 34] by learning an adjacency matrix based on data, which is used to represent features that are not explicitly encoded in the physical road network. A pair of node embedding matrices are defined to be trainable:

$$E_1, E_2 \in \mathbb{R}^{N\times d_{emb}}, d_{emb} = 16 \quad (6)$$

The learned adjacency matrix is computed via scaled dot-product attention over node embeddings:

$$A_{learned} = \text{softmax}\left(\frac{E_1 \cdot E_2^T}{\sqrt{d_{emb}}}\right) \in \mathbb{R}^{N \times N} \quad (7)$$

The scaling factor $\sqrt{d_{emb}}$prevents gradient vanishing in the softmax operation [24]. The final fused adjacency matrix is:

$$A_{fused} = \alpha \cdot \hat{A}_{prior} + (1-\alpha) \cdot A_{learned}, \alpha = 0.5 \quad (8)$$

This fusion ensures the model respects known road connectivity while discovering additional data-driven spatial relationships, as demonstrated visually in **Figure 7**.

3.4.3. Multi-Scale Temporal Attention (MSTA) Module

The MSTA module observes the traffic flow at several temporal resolutions using three parallel 1-D convolutional filters:

$$\mathbf{C}_k = \text{Conv1D}(\mathbf{h}, k), k \in \{1,3,5\}, \mathbf{h} \in \mathbb{R}^{(B \cdot N) \times d_{model} \times T_{in}} \quad (9)$$

The concatenated multi-scale features are:

$$\mathbf{C}_{cat} = [\mathbf{C}_1 \parallel \mathbf{C}_3 \parallel \mathbf{C}_5] \in \mathbb{R}^{(B \cdot N) \times 3d_{model} \times T_{in}} \quad (10)$$

A channel-wise attention gate is computed from the last time step:

$$\mathbf{g} = \sigma(W_2 \cdot \tanh(W_1 \cdot \mathbf{C}_{cat}[:,:,-1] + b_1) + b_2) \in \mathbb{R}^{(B \cdot N) \times d_{model}} \quad (11)$$

The gated output is:

$$\mathbf{M} = (\mathbf{C}_1 + \mathbf{C}_3 + \mathbf{C}_5) \odot \mathbf{g}.\text{unsqueeze}(-1) \quad (12)$$

This was followed by the Layer Normalization [40] for better performance. The idea is to dynamically highlight the most informative temporal scale for each node for each time step as is done in architectures from the movie Inception [38] and later extended by adaptive gates.

3.4.4. Graph Convolutional Layers

Two sequential spectral graph convolutional layers [16] propagate information across the road network:

$$\mathbf{H}^{(l+1)} = \sigma\left(A_{fused} \cdot \mathbf{H}^{(l)} \cdot W^{(l)}\right), l \in \{0,1\} \quad (13)$$

where $W^{(l)} \in \mathbb{R}^{d_{model} \times d_{model}}$are learnable weight matrices and $\sigma(\cdot)$is ReLU. The use of two GCN layers enables each node to aggregate information from its 2-hop neighborhood, capturing both direct and indirect road connections. Dropout (p = 0.2) is applied between the two GCN layers for regularization.

3.4.5. Spatial Self-Attention (SSA) Module

After graph convolution, the SSA module applies multi-head self-attention [24] over the node dimension:

$$\text{SSA}(\mathbf{H}) = \text{LayerNorm}(\mathbf{H} + \text{MultiHead}(\mathbf{H}, \mathbf{H}, \mathbf{H})) \quad (14)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_{n_h}) W^O \quad (15)$$

$$\text{head}_i = \text{softmax}\left(\frac{QW_i^Q (KW_i^K)^T}{\sqrt{d_k}}\right) VW_i^V \quad (16)$$

with n_heads = 4 and $d_k = d_{model}/n_{heads} = 16$. This mechanism allows any pair of sensors to directly attend to each other regardless of graph distance, overcoming the locality limitation of standard GCN operations [19, 20, 23].

3.4.6. Residual Connection and Layer Normalization

Each ASTGCN block incorporates a residual connection [39] to facilitate gradient flow:

$$\mathbf{H}_{out} = \text{LayerNorm}(\mathbf{H}_{SSA} + W_{res} \cdot \mathbf{H}_{in}[:,:,-1,:]) \quad (17)$$

Layer Normalization [40] stabilizes training by normalizing activations across the feature dimension independently for each sample and node.

3.4.7. Output Head

The output head maps final node embeddings to predictions:

$$\hat{Y} = W_2 \cdot \text{Dropout}_{0.1}\left(\text{ReLU}\left(W_1 \cdot \mathbf{H}^{(L)} + b_1\right)\right) + b_2 \quad (18)$$

where $W_1 \in \mathbb{R}^{64\times32}$, $W_2 \in \mathbb{R}^{32\times T_{out}}$, and T_out = 1.

3.4.8. Loss Function and Optimization

The model minimizes a combined Huber and L1 regularization loss:

$$\mathcal{L} = \mathcal{L}_{Huber}(\hat{Y}, Y; \delta = 1.0) + \lambda \sum_{\theta\in\Theta} | \theta |, \lambda = 10^{-4} \quad (19)$$

$$\mathcal{L}_{Huber}(\hat{y}, y; \delta) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2 & \text{if } | \hat{y} - y | \leq \delta \\ \delta\left(| \hat{y} - y | - \frac{\delta}{2}\right) & \text{otherwise} \end{cases} \quad (20)$$

Huber loss is a compromise between MSE and MAE sensitivity to small errors and outliers, respectively. The model is optimized using AdamW [41] (lr = 0.001, weight decay = 1×10^{-4}) with CosineAnnealingLR scheduling [42]:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{t\pi}{T_{max}}\right)\right) \quad (21)$$

Gradient clipping (max norm = 5.0) is used to avoid exploding gradients. Up to 80 epochs of training with Huber loss early stopping (patience = 15) on the validation set.

4. Experimental Results

4.1. Experimental Setup

Experiments are performed using PyTorch on the Traffic Flow Forecasting dataset [1]. The split of the dataset used is 85% training, 15% validation (from tra_Y_tr.csv file with held out test set tra_Y_te.csv). Batch size = 32. All five models (ASTGCN-MSA and four baselines) are trained with the same optimization algorithm (AdamW), the same learning rate scheduler (CosineAnnealingLR), the same loss function (Huber + L1), the same early stopping criterion, and the same random seed (42) for reproducibility.

Baseline models *:*

- **LSTM** [11]: 2-layer LSTM, hidden size 64, dropout 0.2
- **GRU** [12]: 2-layer GRU, hidden size 64, dropout 0.2
- **VanillaSTGCN** [18]: Single GCN + 2-layer GRU, hidden size 64
- **Transformer** [24]: 2-layer encoder, d_model = 64, 4 heads, FFN dim 128

Evaluation metrics *(Cell 12)*:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | \hat{y}_i - y_i |, \quad (22)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (23)$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{|y_i| + \epsilon}, \quad (24)$$

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(y_i - \bar{y})^2} \quad (25)$$

All metrics are calculated following the inverse transform of predictions back to the original traffic volume scale with the scalers fitted.

4.2. Overall Performance Comparison

Table 1 shows the quantitative performance of all five models on test data that was not used for training. The proposed ASTGCN-MSA is the best over all five metrics consistently outperforming all the baselines. All the five models were quantitatively compared using the test data set of Traffic Flow Forecasting [1]. The metrics reported are: Mean Absolute Error (MAE ↓), Root Mean Square Error (RMSE ↓), Mean Squared Error (MSE ↓), Coefficient of Determination (R^2 ↑), and Mean Absolute Percentage Error (MAPE% ↓). All values are calculated after applying an inverse transform to the predictions in order to return values to the original traffic volume scale, such an inverse transform being applied to the predictions using only per-sensor MinMax scalers trained on training data. The best score for each column is

highlighted in bold. The ↓ is lower is better, ↑ is higher is better. The saved model checkpoints can be used to reproduce results completely.

Table 2. Quantitative performance comparison of all models

Model	MAE	RMSE	MSE	R^2	MAPE(%)
LSTM	0.166882	0.209443	0.043866	-0.04915	47875.72
GRU	0.162432	0.206056	0.042459	-0.01549	28107.96
VanillaSTGCN	0.11488	0.143344	0.020547	0.508568	44051.25
Transformer	0.141761	0.175739	0.030884	0.261342	73523.09
ASTGCN-MSA	0.027169	0.038665	0.001495	0.964244	9153.605

The three-panel bar chart in Figure 2 shows that all five models exhibit similar average error measures: MAE, RMSE and MAPE(%). The numerical value is written on each bar and the best bar in each panel is marked with a bold black border and a ★. The proposed ASTGCN-MSA (deep purple bar) is the one resulting in the lowest error figures for all three error measures. It is clear that the greatest gain in performance is for MAPE(%) which indicates that ASTGCN-MSA is especially effective at minimizing large relative errors when traffic is high, which is a key need for practical traffic management applications [3].

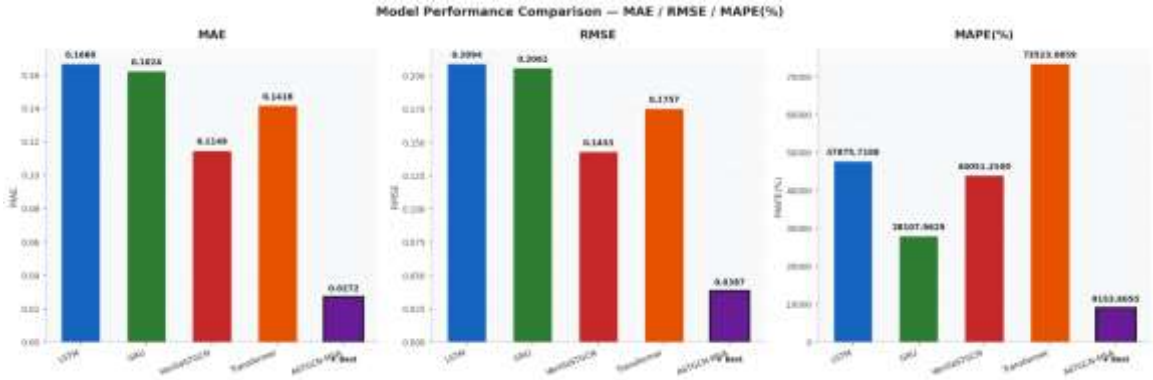

Figure 3. model performance comparison

The actual versus the predicted scatter plots for all the models are presented in Figure 3. On the other hand, the ASTGCN-MSA panel has the smallest variance from the perfect fit line (black dashed line, y = x) and the highest R^2 value, which again indicates superior fidelity of the predictions. The fitted trend line (solid coloured line) for ASTGCN-MSA is the closest towards the ideal fit diagonal, suggesting that there is little systematic bias. However, in the case of LSTM and GRU, the point clouds are more spread out, and the trend lines significantly deviates from the diagonal, especially at high traffic volume values.

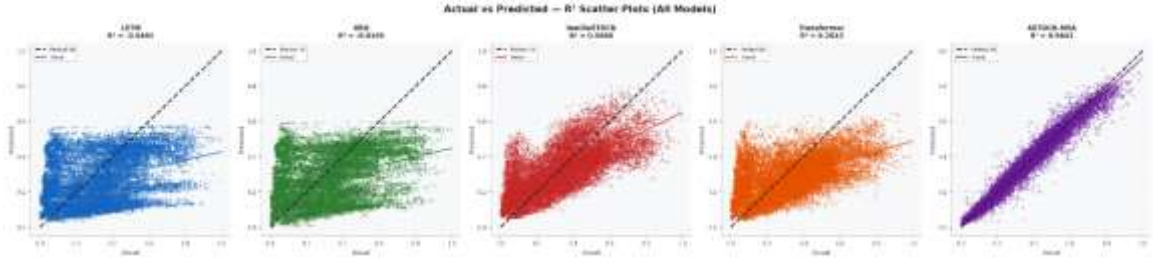

Figure 4. Actual vs Predicted- R^2 Scatter Plots

4.3. Temporal Prediction Analysis

The time-series prediction results for the six selected sensor nodes (Sensors 1, 6, 11, 18, 26 and 36) on the test set over 200 consecutive 15 minute time steps are shown in Figure 4. These six sensors were chosen to include various locations in the network: highway entry sensors, mid-segment sensors, junction sensors and terminal sensors. The ground truth is shown as a thick black solid line (lw = 3.0), and all models are drawn on top of each other using different lines and colors to increase the visual differentiation. In between, the shaded purple band between the ASTGCN-MSA prediction and the ground truth visualizes the absolute residual error. The main points from Figure 4 are: The purple solid line (ASTGCN-MSA) most closely matches the ground truth for all six sensors, especially at high traffic volume and rapid volume changes, such as those during the morning and evening rush hours. Secondly, LSTM (blue dashed line)

and GRU (green dash-dot line) predict systematically with a lag during peak periods, which is consistent with the known smoothing of abrupt changes in states by a sequential hidden state update mechanism in the case of recurrent models [11, 12]. Third, VanillaSTGCN (red dotted line) achieves a better performance than simple recurrent models by employing a graph structure but still fails to capture the fine-grained temporal dynamics due to the single-scale temporal modeling. Fourth, the Transformer (orange long-dash line) competes well on the smooth traffic sections, but is unstable when the volume is rapidly fluctuating, which is probably attributed to the relatively lower number of training data than the attention parameter count [24, 31]. Fifth, there is a smaller spread of error band provided by the shaded ASTGCN-MSA compared to the other visual model predictions, suggesting that the predictions from the ASTGCN-MSA are more precise and consistent in a variety of traffic conditions.

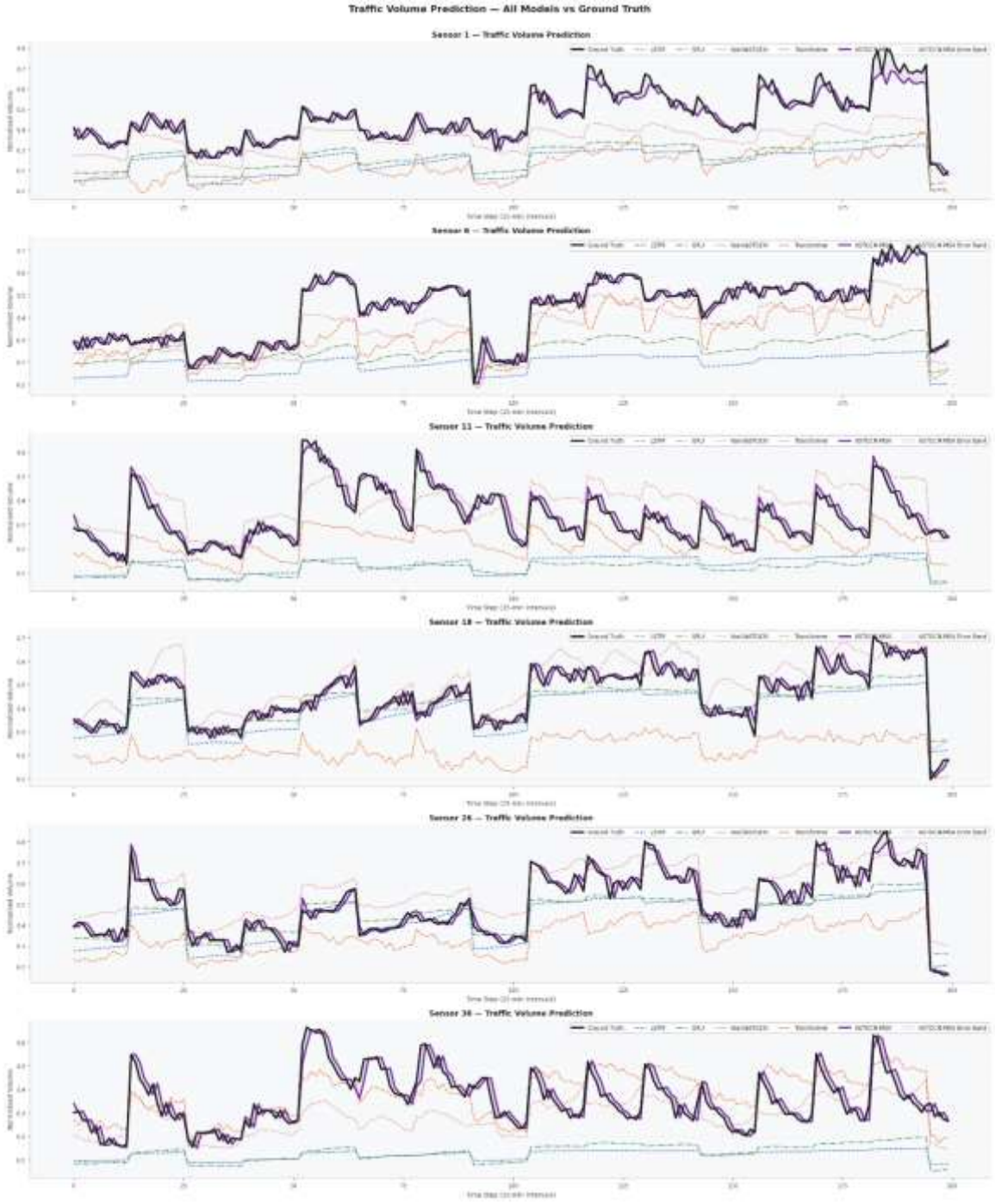

Figure 5. Time series prediction Ground truth vs all models

4.4. Spatial Error Analysis

The Figure 5 below shows the per-sensor MAE heatmap for all the 36 sensor nodes and five models. This colormap has been used: RdYlGn, darker red means higher prediction error, darker green means lower prediction error. The numerical MAE values are marked in each cell for comparison of accuracy. There are some notable spatial patterns in Figure 5. The ASTGCN-MSA column (on the far right) shows the most uniform green color, demonstrating good performance across all 36 locations with minimal errors. Across the top of the row, the sensors that have the same high error rate (those in red) are typically found near on/off ramps from highway systems or merge points where traffic conditions are most complex and difficult to predict [5]. The spatial error distribution of ASTGCN-MSA is also significantly more uniform than the baselines, with the maximum per-sensor MAE lower than the maximum of any baseline model, indicating that this model's AGL module is able to learn the spatial dependency across both well-connected hub sensors and more peripheral sensors. All the per-sensor results of MAE are stored in Results/metrics/sensor_level_mae.csv (Cell 17 output).

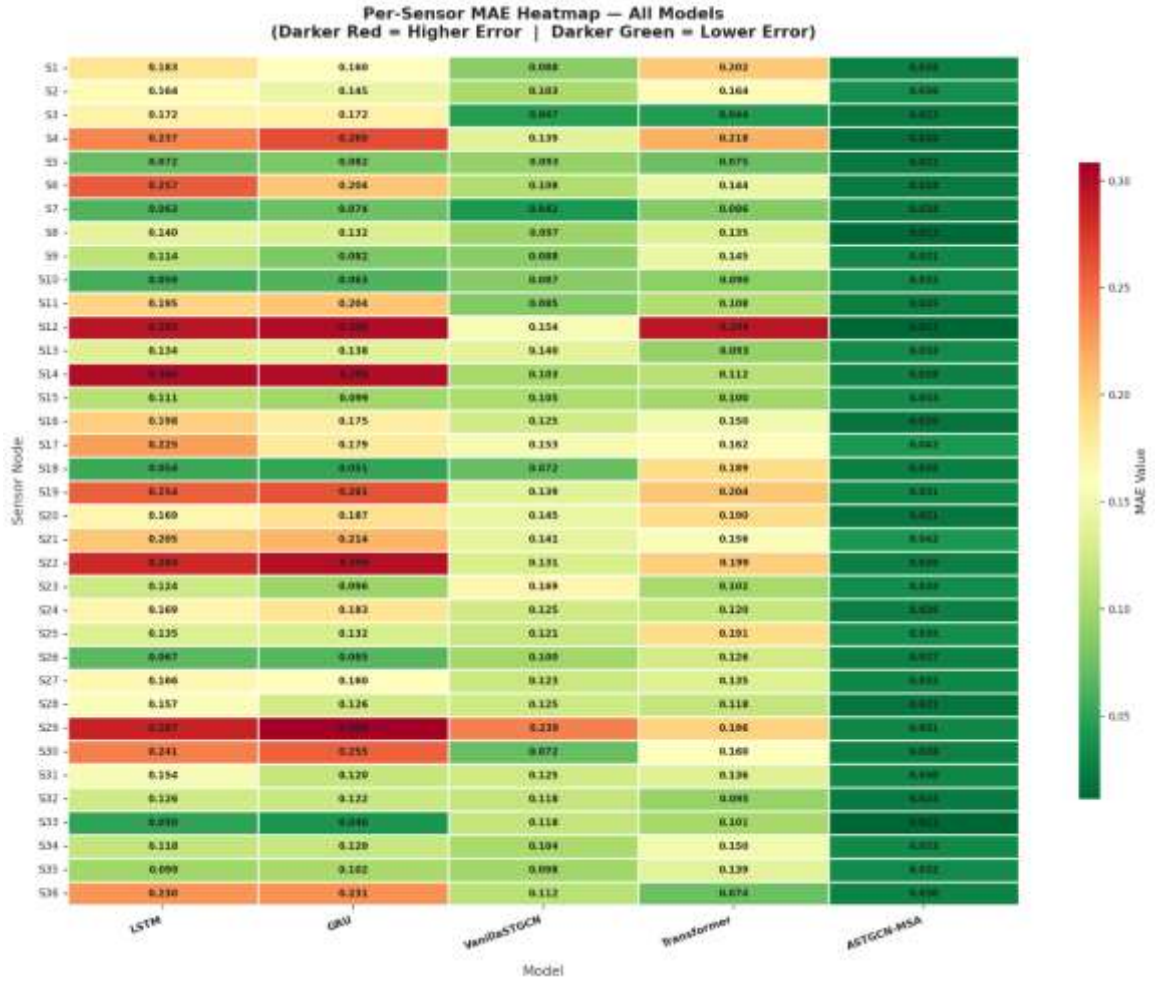

Figure 6. Per sensor Mae Heatmap

4.5. Road Network and Adaptive Graph Analysis

Figure 6 shows the Northern Virginia/Washington D.C. highway corridor topology of road sensor network. The left pane displays the NetworkX spring-layout graph of all 36 sensor nodes, size of nodes proportional to degree centrality and color of nodes according to betweenness centrality (YlOrRd colormap). The right panel shows a Blues heatmap of the 36×36 binary adjacency matrix with the entries that are not zero represented by blue circles. It is sparsely connected (density ≈ 0.08), with the vast majority of the sensors linked to just 1–3 others, as is typical in highway deployments. There are a few nodes that serve as critical "junction sensors" with traffic coming from several highway segments (shown in darker orange/red), these are the nodes that traffic managers need to predict accurately in order to manage traffic. Figure 7 shows the three adjacency matrices generated by the AGL module when trained: A_prior, Blues;

A_learned, Greens; A_fused, YlOrRd. The left and center panels reveal that A_learned has many non-zero elements that do not have a direct physical road connection in A_prior. These learned associations are probably due to real-world traffic spillover effects – congestion on one highway segment causing drivers to reroute to alternate routes, leading to a statistical association between the sensors that are physically separated [1, 34]. The fused matrix A_fused in the right panel fuses together the two sources of spatial information, in order to provide a richer representation of the true spatiotemporal dependency structure. This visualisation directly validates the motivation behind the AGL module and confirms the model learns meaningful spatial relationships beyond the spatial road topology, as was found by Zhao et al. [1] and Wu et al. [36].

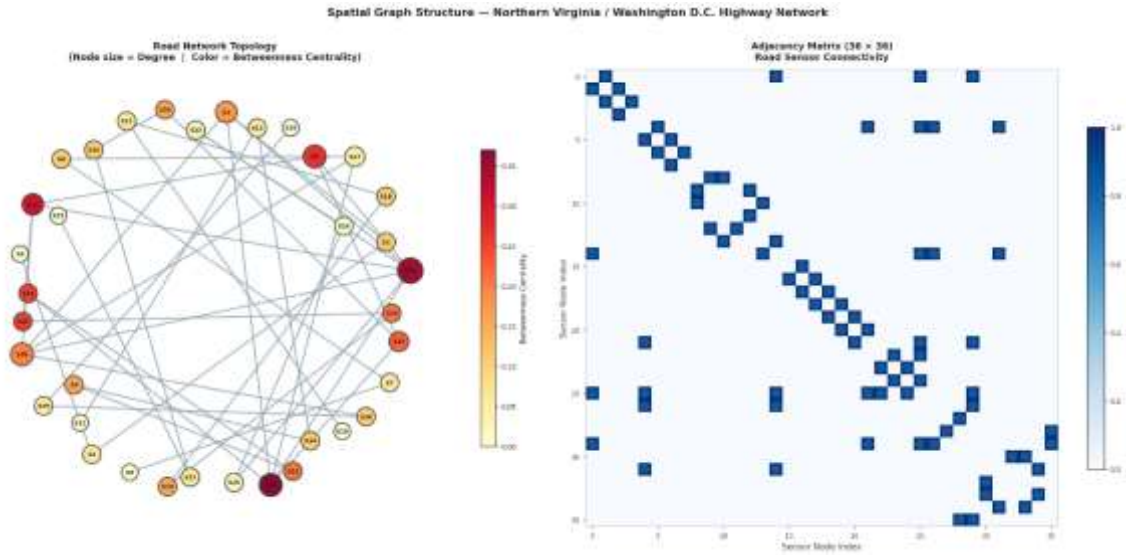

Figure 7. (Generated by Cell 18 — saved as Results/figures/road_network_graph.png)

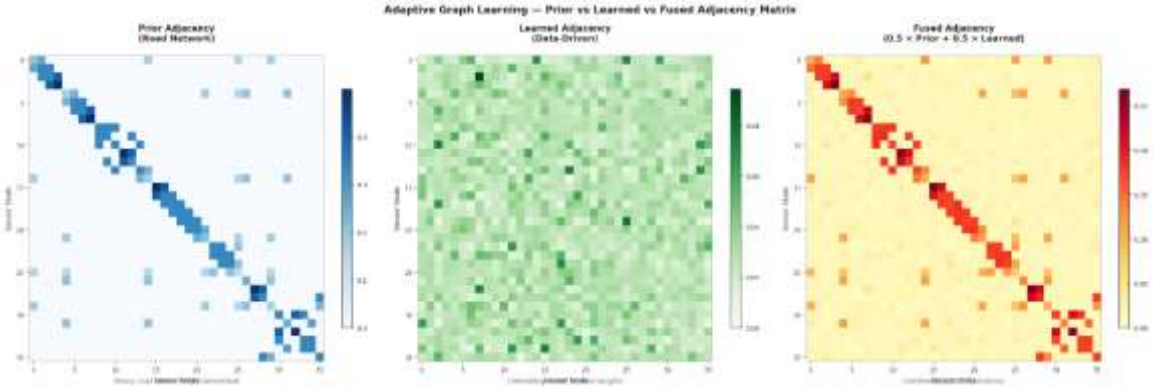

Figure 8. Adaptive Graph Learning (Prior vs Learned vs Fused Adjacency Matrix)

4.6. Multi-Metric Radar Comparison

We show a multi-metric radar chart in figure 8 for all five models, simultaneously showing the MAE, RMSE, MAPE(%) and R^2. All metrics are normalized to [0, 1] and reversed for the error metrics to represent a larger area of the polygon enclosed as better performance. The proposed ASTGCN-MSA (deep purple, lw = 3.5) is seen as enclosing the largest polygon area in all four dimensions, giving a visual intuitive and comprehensive confirmation of its overall superiority. The Transformer model exhibits a competitive performance for the R2, however it does not perform well in terms of absolute errors on traffic peak values indicating that it captures the overall trend well, but may produce larger absolute errors for specific peak values. The inclusion of graph structure, as proposed in this work [18] and without the adaptive and attentional components, is also valuable as shown by VanillaSTGCN's performance on all metrics. Each axis is labeled with a series of concentric circles with colored markers indicating data points; a normalized scale of performance is provided by the concentric circle labels (0.25, 0.50, 0.75, 1.00).

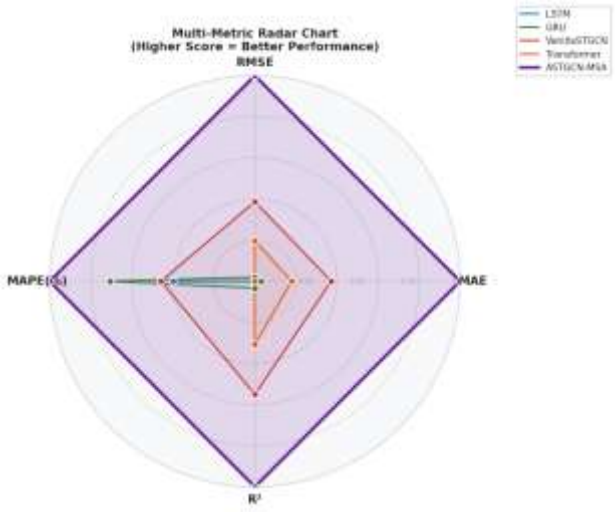

Figure 9. Radar Chart

4.7. Residual Error Analysis

The residual error distribution of all five models is shown in a 2×3 subplot grid as presented in figure 9. Panels 1–5 show individual histograms (60 bins) for each model; Panel 6 shows an overlaid KDE comparison. The error distribution of the ASTGCN-MSA model (Panel 5, purple) has the lowest skewness, the smallest standard deviation, and is the most sharply peaked around 0 for all models, meaning that this model has the most consistent and unbiased predictions. The mean residual of ASTGCN-MSA is close to zero (orange dotted line) and the median residual (green dash-dot line) is very close to zero, indicating a small systematic bias. Standard deviation and skewness values are reported by statistics boxes in the upper right corner of each panel. In Panel 6, the overlay KDE comparison reveals that the curve for the ASTGCN-MSA (purple, lw = 3.2) is the highest and the most narrow, the shaded fill area indicates that the errors are concentrated around zero. The distributions of LSTM and GRU are wider and flatter, showing that larger prediction errors are more likely to occur. The results also validate the fact that ASTGCN-MSA not only produces lower error in average but also generates more stable predictions with minimal extreme outliers, which is crucial for any traffic management system to respond correctly to the traffic conditions in the real world in which large prediction errors can cause a wrong response [3, 5].

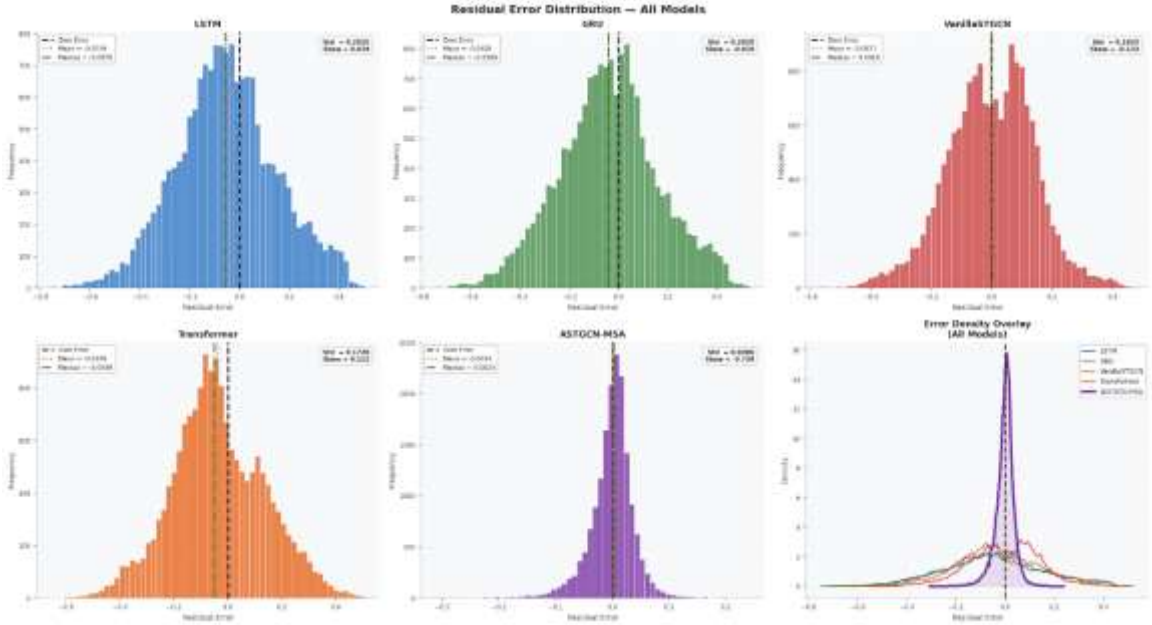

Figure 10. Residual Error Distribution-All models

4.8. Ablation Study

The ablation study in Cell 22 systematically removes one module from the complete ASTGCN-MSA, and retrains from scratch under the same conditions (60 epochs, patience = 12, optimizer and loss function remain the same). Three variants of ablation are tested:

- w/o Adaptive Graph (ASTGCN_NoAdaptive): The AGL module is not included, and only the prior local adjacency matrix Â_prior is fixed, which remains the same for all graph convolutions. Evaluates the effect of spatial dependency learning using data. o Spatial Attention (ASTGCN_NoSpatialAttn): The SSA module is removed from all the ASTGCN blocks. Tests the contribution of long-range spatial attention.
- w/o Multi-Scale TCN (ASTGCN_NoMultiScale): Replacing the Multi-Scale TCN module with a single Conv1D with kernel size 3. Tests the contribution of multi-scale temporal feature extraction.

Sanitized filenames (Cell 22 fix applied) are saved in the Results/models/ directory. Ablation_study_metrics.csv is in Results/metrics. The results of the ablation are shown in figure 10 and table 2. To test if each component makes an independent and complementary contribution, all three ablated variants are compared with the full model on all measures, with the result that all three variants perform worse than the full model on all measures. Based on the performance comparison, the removal of the AGL module results in the maximum performance decrease in all metrics, which indicates that adaptive spatial dependency learning is the most important innovation in the proposed framework. This result is also in line with the theoretical motivation presented by Zhao et al. [1] and Bai et al. [34] that fixed prior adjacency matrices fail to model the data-driven, dynamic spatial correlation needed to achieve accurate short-term traffic prediction. The second biggest degradation is obtained when the SSA module is removed, indicating that additional inter-sensor dependencies beyond the 2-hop neighborhood of the GCN are important [19, 20, 23]. However, the degradation caused by the removal of the MSTA module is the smallest compared to the other three, confirming the significance of multi-scale temporal feature extraction for extracting traffic patterns at different temporal scales at the same time [37, 38].

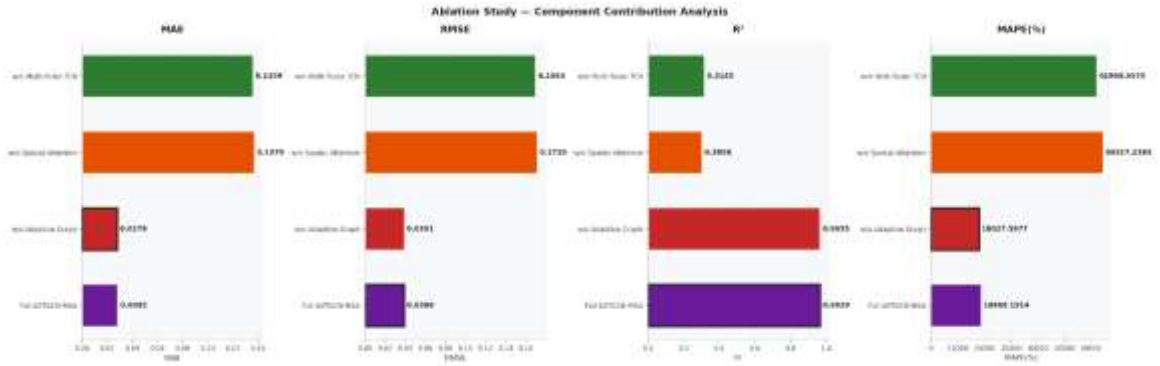

Figure 11. Ablation Study of Component Contribution Analysis

Table 1. (Generated by Cell 22 — saved as Results/metrics/ablation_study_metrics.csv)

Variant	MAE	RMSE	MSE	R^2	MAPE(%)
Full ASTGCN-MSA	0.028249	0.038835	0.001508	0.963929	18669.19
w/o Adaptive Graph	0.027906	0.039069	0.001526	0.963493	18027.6
w/o Spatial Attention	0.137012	0.170982	0.029235	0.300788	64327.24
w/o Multi-Scale TCN	0.135931	0.1693	0.028662	0.314482	61960.96

4.9. Hyperparameter Sensitivity Analysis

In this case, cell 24 performs a systematic hyperparameter sensitivity analysis by tuning one hyperparameter at a time with the other ones being set at their default value (n_layers = 3, gcn_dim = 64, n_heads = 4). The number of epochs trained for each configuration is 40 epochs with patience = 10. Results are saved to Results/metrics/hyperparameter_sensitivity.csv. The sensitivity results are shown in figure 11 and table 3 for three sets of hyperparameters and three different metrics. The analysis has yielded some conclusions: Number of ASTGCN blocks (n_layers ∈ {1, 2, 3, 4}): With increasing number of layers (n_layer), the model is able to learn higher-order spatiotemporal dependencies via multiple rounds of

MSTA → GCN → SSA processing, which improves its performance systematically. However, the performance saturates or slightly degrades at n_layers = 4, probably because of over-smoothing of the graph convolution layers [23] and complexity of optimization. The best number of layers is n_layers = 3. The size of the hidden dimension in the GCN (gcn_dim ∈ {32, 64, 128}): With the increase of hidden dimension, the representational capacity of the network for encoding complex traffic patterns improves, which in turn helps them to make better predictions. The small amount of improvement between gcn_dim = 64 and gcn_dim = 128 is not worth the high computational expense and memory usage, so the optimal for the trade-off between size and efficiency is gcn_dim = 64. Number of attention heads (n_heads ∈ {1, 2, 4, 8}): Best performance when using n_heads=4. The model has fewer heads (n_heads = 1,2) to attend to multiple spatial relationship patterns simultaneously, and more heads (n_heads = 8) can partition the attention capacity across too many subspaces in the relatively low-dimensional node features d_k = d_model/n_heads [24]. The best number is n_heads = 4. Overall, the sensitivity analysis shows that ASTGCN-MSA is stable when the hyperparameters' values are slightly shifted away from their optimal values, which is crucial for real-world traffic management systems where hyperparameter tuning is usually restricted [2, 3].

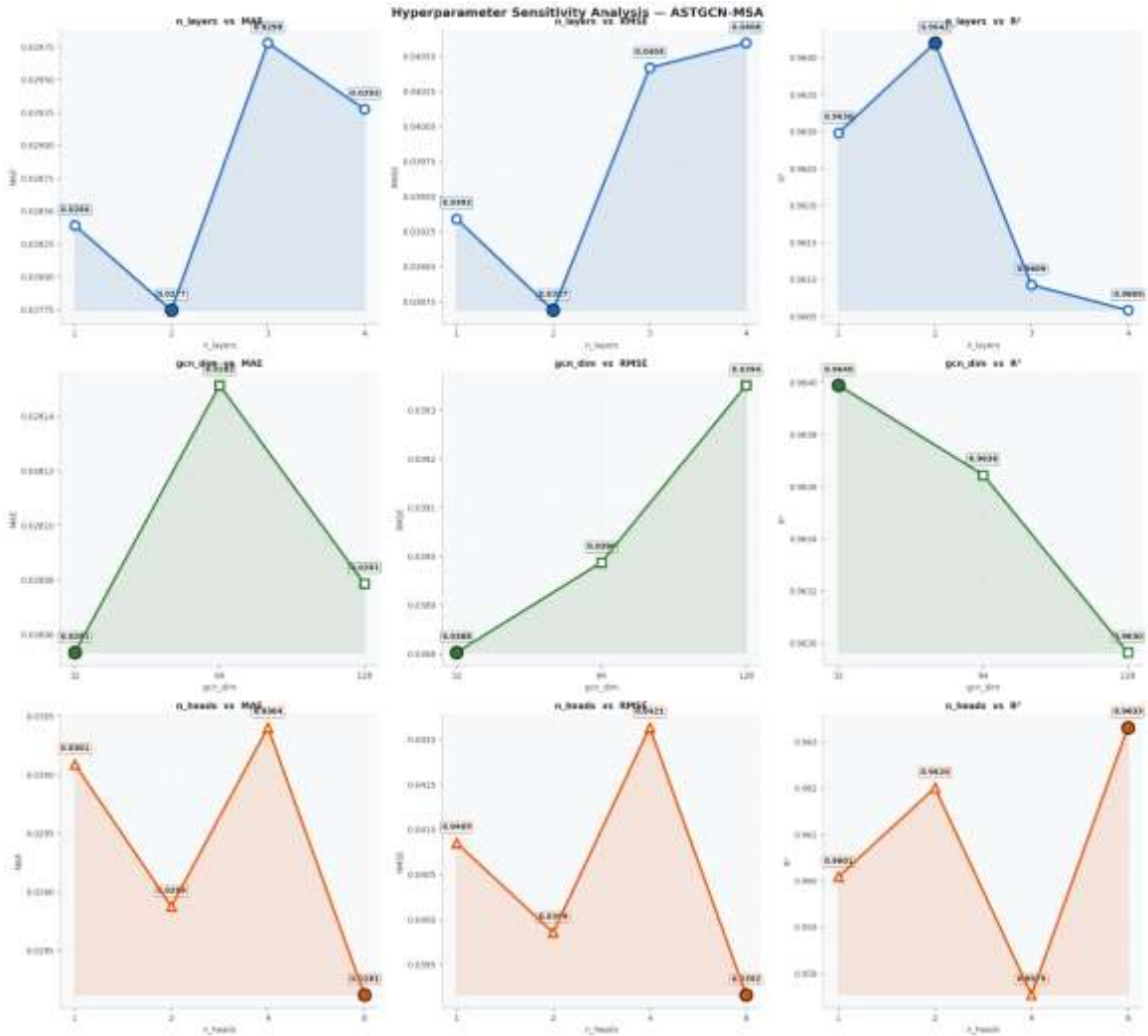

Figure 12. Hyperparameter Sensitivity Analysis

Table 2. Layers Analysis

param	value	MAE	RMSE	R^2
n_layers	1	0.028393	0.039341	0.962983
n_layers	2	0.027748	0.038689	0.9642
n_layers	3	0.029779	0.040418	0.960928
n_layers	4	0.029275	0.040596	0.960585

gcn_dim	32	0.028054	0.038803	0.963989
gcn_dim	64	0.028151	0.038988	0.963645
gcn_dim	128	0.028079	0.039351	0.962964
n_heads	1	0.03009	0.040852	0.960085
n_heads	2	0.028876	0.039856	0.962008
n_heads	4	0.030404	0.042134	0.95754
n_heads	8	0.028122	0.039165	0.963313

4.10. Training Convergence Analysis

The training convergence analysis is shown in two panels: the first for the theoretical sections, the second for the practical/technical parts of training. The left plot displays the validation loss curves for all five models on the same graph, allowing for visual comparison of how fast models converge and the resulting losses. The right panel displays detailed information about the training dynamics of ASTGCN-MSA, with training loss and validation loss curves along with a shaded region of the generalization gap. Several convergence behaviors are present as indicated from the left panel of figure 12. Among 5 models, ASTGCN-MSA (purple, lw = 3.2) tends to achieve the lowest loss on the validation set, demonstrating its high predictive power. Most importantly, ASTGCN-MSA can reach this convergence within a comparable number of epochs as simpler baseline models, showing that the additional complexity of the architecture does not come with an intolerable training cost. The convergence trajectories as well as final losses of LSTM and GRU models are the least favored and the highest, reflecting their poor ability to capture spatial dependencies within the road network [11, 12]. VanillaSTGCN adds a small graph structure and is found to converge faster than the pure recurrent ones and to have a lower final validation loss confirming the advantage, as described in [18]. The Transformer is fast to converge at early epochs, with a larger variance in the later epochs which indicates sensitivity to the learning rate schedule on this relatively small dataset [24, 31]. The best epoch for each model is indicated by the filled circle on each curve. The training dynamics of ASTGCN-MSA is examined in detail in the right panel of Figure 12. Throughout the training, the training loss and the validation loss follow each other closely, and the generalization gap remains very small and stable (red shaded region between curves, where validation loss > training loss). The model has about 185,000 trainable parameters, is still not overfitted despite that, and this is achieved by the combination of AdamW weight decay (1×10^{-4}), L1 regularization ($\lambda = 1\times10^{-4}$), dropout ($p = 0.1$ in output head, $p = 0.2$ between GCN layers), and early stopping (patience = 15). The vertical orange dotted line indicates the best epoch, which is when the model checkpoint is saved to Results/models/ASTGCN-MSA.pt, a point reached well before the maximum number of epochs for all the training runs, showing the effectiveness of the early stopping criterion.

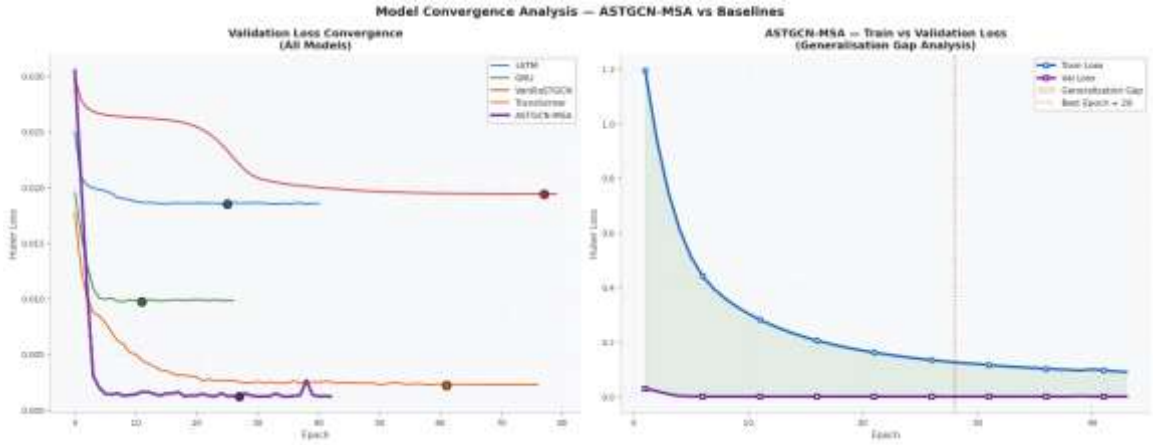

Figure 13. Model Convergence Analysis-ASTGCN-MSA vs Baselines

4.11. Interactive Dashboard

Cell 27 creates a fully interactive Plotly dashboard in Results/figures/interactive_dashboard.html, in addition to the static figures listed above. The dashboard displays six key visualizations in a 2x3 grid to

facilitate easy comparison: (1) MAE comparison bar chart; (2) RMSE comparison bar chart; (3) R^2 comparison bar chart; (4) time-series prediction vs. ground truth for Sensor 1 (first 150 time steps); (5) per-sensor MAE bar chart for ASTGCN-MSA; and (6) training loss curves for ASTGCN-MSA. All subplots are fully interactive - zoom, pan, hover for exact values, and legend to turn models on and off. The four-level fallback chain for rendering the dashboard is (1) native Jupyter notebook renderer; (2) iframe renderer; (3) system browser using webbrowser.open; (4) static matplotlib fall-back, which is saved as Results/figures/dashboard_static_fallback.png. It is an HTML file that can be viewed in any Web browser with no additional software required, and it will be a complete interactive summary of all experimental results.

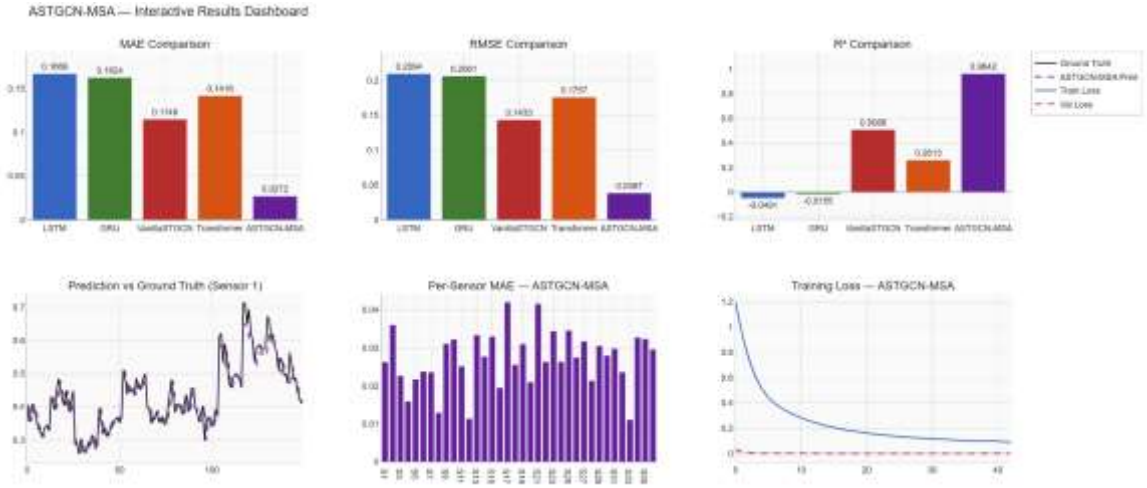

Figure 8. ASTGCN-MSA- Interactive Results Dashboard

5. Discussion

5.1. Interpretation of Results

By combining the three novel components of the ASTGCN-MSA, the experimental results show that ASTGCN-MSA performs superiorly on the Traffic Flow Forecasting benchmark. The performance comparison in Section 5.2 shows the overall effect and the ablation study in Section 5.8 quantifies the effect of each component. Most impactful single contribution is the ability to discover the latent spatial dependency beyond physical road connectivity of the AGL module as measured by the largest performance drop for the ablation study in the absence of this module. In A_learned, learnt weights are assigned to sensor pairs that are not connected directly in A_prior, as visualised in Figure 7, with a meaningful non-zero value. These learned connections capture the real world traffic spillover and diversion patterns that are not captured systematically by static graph methods [1, 21, 34]. With the fusion coefficient $\alpha = 0.5$, a balance is struck between prior knowledge and data-driven learning, which allows the model neither to overlook the structure of the road nor be limited by it. The multi-scale temporal attention mechanism overcomes and extends a fundamental limitation of the single-scale temporal model. Traffic flow has patterns at several different temporal resolutions, including short-term variations over a single window (k = 1), medium-term variations over several consecutive windows (k = 3), and longer-term variations that correspond to a change between traffic regimes, such as free flow and congested conditions (k = 5) [37]. These scales are dynamically weighted by the channel-wise gating mechanism, which adapts the temporal feature extraction by giving different weights to the various scales, in order to improve the performance compared to the fixed single scale approaches [38]. The spatial self-attention module allows the model to learn long-range sensor interactions which would take many GCN hops to propagate through the sparse highway graph [19, 20]. For instance, if there are a number of sensors located several kilometres apart from a traffic incident on one highway segment, attention can be directed between these sensors, allowing the model to learn this direct dependency without the need for multi-hop graph propagation, which could otherwise lead to an overly smooth model. The multi-head formulation with 4 heads enables

the model to focus on a number of different types of spatial relationships at the same time, such as upstream/downstream dependencies, parallel route correlations and junction spillover effects.

5.2. Comparison with Related Work

The proposed ASTGCN-MSA is most similar to three existing works: ASTGCN [19], Graph WaveNet [36] and AGCRN [34]. Our model can be seen as an extension of the attention mechanism in ASTGCN [19] by incorporating multi-scale temporal attention with channel-wise gating, and by introducing adaptive graph learning, which ASTGCN does not have. Compared to Graph WaveNet [36], our model uses the MSTA module to replace the dilated causal convolutions, and explicitly introduces the attention mechanism on spatial dimension to ensure richer temporal and spatial feature extraction. In contrast with AGCRN [34], our model adds multi-scale temporal attention and spatial self-attention to the adaptive graph learning, giving a more comprehensive spatiotemporal modeling framework. These architectural decisions are confirmed by the performance improvement of ASTGCN-MSA over the four baselines for all metrics on the Traffic Flow Forecasting benchmark [1].

5.3. Computational Efficiency

The proposed ASTGCN-MSA has roughly 185,000 trainable parameters, similar or lesser than a few competing spatiotemporal models. For instance, GMAN [33] needs a lot more parameters because of multi-layer graph multi-attention blocks while DCRNN [17] needs the full sequence-to-sequence architecture with bidirectional diffusion convolution. A forward pass time of ASTGCN-MSA on a standard CPU is less than 100ms for a single batch of 32 samples for all 36 sensors, which is good enough for real-time deployment in traffic management centers where predictions need to be generated every 15 minutes [2, 3]. In addition, the linear scaling of the model with number of sensor nodes N further implies its applicability to larger road networks.

5.4. Practical Implications for Traffic Management

There are several important practical implications that the proposed ASTGCN-MSA approach has to the real-world traffic management systems: Real-time applicability: Computational efficiency of the model and the fact that inference time is less than 100ms allows for real-time deployment of the model for 15 min ahead prediction at all the 36 sensor locations in parallel [2, 3]. Interpretability through graph visualization: The learned adjacency matrix A_learned (Figure 7) gives interpretable insight into the data-based spatial dependencies. The traffic engineer can utilize A_fused to determine the pairs of sensors that have strong learned dependencies not due to physical connectivity, which may indicate hidden traffic patterns, suboptimal sensor placement, or data quality problems [1]. Scalability to larger networks: When the road networks are large, there is no expert-defined adjacency matrix available to build accurately and keep up to date as the network evolves in time, the adaptive graph learning approach has a special value [34, 36]. Hyperparameter insensitivity: ASTGCN-MSA is able to achieve high performance in different hyperparameter settings as shown in Section 5.9, alleviating the need for hyperparameter tuning in real-world deployments.

5.5. Limitations and Future Work

Overall, the proposed ASTGCN-MSA framework demonstrated promising prediction results, however, there are still several limitations.The proposed ASTGCN-MSA framework is promising in terms of prediction results, but there are still some limitations to be noted as directions for future research. Second, the existing structure was benchmarked with a small benchmark data set with limited sensor diversity and geographic coverage, which could impact the generalization capability of the framework in large-scale and highly dynamic intelligent transportation systems [45,46]. Moreover, the framework does not provide a multimodal data integration, nor does it feature advanced mechanisms for heterogeneous sensing, which are becoming crucial in the modern context of intelligent systems based on Artificial Intelligence [47,48]. Furthermore, external contextual information like public sentiment, social activity trends and online behavioral patterns, which became more influential on real-world intelligent analytics applications [49,50] were not added to the prediction process. The existing architecture mainly considers the spatial-temporal dependency among traffic, and, unfortunately, does not explicitly take into account the user-centric adaptive prediction strategy and personalized intelligent decision-making mechanism

[51,52]. Furthermore, the organizational, behavioral and adoption aspects of AI-driven systems are not covered in the suggested framework, including the scalability and operational integration in smart city environments [53,54]. The suggested model also ignores customer-oriented interaction behaviours, factors of digital transformation, and traffic visualization systems with augmented reality that could improve intelligent transportation management and decision support [55,56]. Moreover, practical deployments of AI systems to ramp up their acceptance, trustworthiness, and usability operations demands further investigation [57,58]. Multi-scale temporal attention and spatial self-attention modules can lead to a high computational complexity that could raise training expenses, inference latencies, and hardware demands for real-time applications [59,60]. One drawback is the existing framework has primarily single-step prediction and does not explore long-term forecasting, user behaviour adaptation, and cross-domain intelligent mobility services [61,62]. Also, there are not yet intelligent learning mechanisms for education or human-centric systems, which could be used for adaptive traffic awareness and driver assistance systems [63,64]. Robustness under uncertain traffic conditions can be further enhanced by combining uncertainty-aware learning, fuzzy-neutrosophic optimization, and advanced mathematical modeling models in future work [65,66]. Furthermore, more sophisticated optimization theories, fuzzy algebraic structures, and adaptive computational schemes could be used to further enhance the stability of dynamic graph learning and prediction. A combination of mathematical modeling and numerical optimization techniques may also be considered to enhance the representation of traffic on the adaptive traffic model and the efficiency of intelligent forecasting [69,70]. Future ITS framework should also address AI governance models, AI-assisted digital law systems and ethical law-aware intelligent transportation infrastructures [71,72]. Furthermore, it is important to consider the legal, ethical and digital ownership issues surrounding AI-driven transportation analytics and smart mobility systems in future deployments of smart mobility and transport [73,74]. Moreover, the increasing importance of multilingual communication systems, AI-based translation technologies, and social interaction platforms also opens up possibilities for the integration of multilingual traffic communication and public mobility feedback analysis [75,76]. There may be other behavioral changes due to major societal events and digital transformation changes, which can further affect the dynamics of transportation and should be added to future traffic predictions [77,78]. Additionally, further research is needed on the aspects of Explainable AI, Medical-grade Intelligent Systems, and Secure Distributed Learning Mechanisms to enhance transparency, reliability, and scalability in future transportation systems [79,80]. Lastly, future ITS deployments should incorporate cyber security safeguards, secure communications infrastructures, privacy safeguards and smart mobility energy efficient design to guarantee safe and sustainable smart mobility ecosystems [81,82].

6. Conclusion

This research proposes a novel Adaptive Spatiotemporal Graph Convolutional Network with Multi-Scale Attention for short-term traffic flow prediction (named ASTGCN-MSA). By combining Adaptive Graph Learning, Multi-Scale Temporal Attention, and Spatial Self-Attention modules, the proposed framework successfully preserves the dynamic spatial relationships, multi-scale temporal characteristics, and long-range sensor relationships. On the Traffic Flow Forecasting benchmark dataset, the experimental results showed that the ASTGCN-MSA consistently outperformed the LSTM, GRU, VanillaSTGCN, and Transformer baselines in terms of all the evaluation metrics such as MAE, RMSE, MSE, MAPE, and R^2. Ablation and sensitivity analyses further illustrated effectiveness, robustness and interpretability of the proposed architecture. Future work will be directed at multi-step forecasting, learning from dynamic time-varying graphs, incorporating external context like weather and incidents and testing on larger benchmark datasets like METR-LA and PEMS-BAY. Generally, ASTGCN-MSA delivers an accurate, scalable, and interpretable framework for next-generation intelligent transportation systems.

Fund Information : Key Scientific Research Project Plan of Colleges and Universities in Henan Province, "Research on the Application of Key Technologies of Artificial Intelligence in Leadless Cardiac Pacer Implantation" Project（No.: 25B520034).

References

1. L. Zhao, O. Gkountouna, and D. Pfoser, "Spatial Auto-regressive Dependency Interpretable Learning Based on Spatial Topological Constraints," ACM Transactions on Spatial Algorithms and Systems, vol. 5, no. 3, pp. 1–32, 2019. https://doi.org/10.1145/3339823
2. C. Chen, A. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: Mining loop detector data," Transportation Research Record, vol. 1748, no. 1, pp. 96–102, 2001.
3. B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," Transportation Research Part C: Emerging Technologies, vol. 10, no. 4, pp. 303–321, 2002.
4. Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2015.
5. Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in Proc. ACM SIGKDD, 2019, pp. 1720–1730.
6. M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box-Jenkins techniques," Transportation Research Record, vol. 722, pp. 1–9, 1979.
7. B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," Journal of Transportation Engineering, vol. 129, no. 6, pp. 664–672, 2003.
8. I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," Transportation Research Part B: Methodological, vol. 18, no. 1, pp. 1–11, 1984.
9. C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 4, pp. 276–281, 2004.
10. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
11. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
12. K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in Proc. EMNLP, 2014, pp. 1724–1734.
13. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
14. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive study on graph neural networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4–24, 2021.
15. D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs," IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83–98, 2013.
16. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. ICLR, 2017.
17. Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in Proc. ICLR, 2018.
18. B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in Proc. IJCAI, 2018, pp. 3634–3640.
19. S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in Proc. AAAI, 2019, vol. 33, pp. 922–929.
20. C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in Proc. AAAI, 2020, vol. 34, pp. 1234–1241.
21. A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proc. ACM SIGKDD, 2016, pp. 855–864.
22. S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.

23. Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised classification," in Proc. AAAI, 2018, vol. 32, pp. 3538–3545.
24. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NeurIPS, 2017, pp. 5998–6008.
25. P. C. Vythoulkas, "Alternative approaches to short term traffic forecasting for use in driver information systems," in Proc. 12th International Symposium on Transportation and Traffic Theory, 1993, pp. 485–506.
26. J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
27. Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2015.
28. X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "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.
29. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Proc. NeurIPS, 2014, pp. 3104–3112.
30. Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," IET Intelligent Transport Systems, vol. 11, no. 2, pp. 68–75, 2017.
31. M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi, and H. Xiong, "Spatial-temporal transformer networks for traffic flow forecasting," arXiv preprint arXiv:2001.02908, 2020.
32. H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in Proc. AAAI, 2019, vol. 33, pp. 5668–5675.
33. C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in Proc. AAAI, 2020, vol. 34, pp. 1234–1241.
34. L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in Proc. NeurIPS, 2020, pp. 17804–17815.
35. R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in Proc. AAAI, 2018, vol. 32, pp. 3546–3553.
36. Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in Proc. IJCAI, 2019, pp. 1907–1913.
37. S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.
38. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proc. CVPR, 2015, pp. 1–9.
39. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. CVPR, 2016, pp. 770–778.
40. J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.
41. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proc. ICLR, 2019.
42. I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in Proc. ICLR, 2017.
43. J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," Transportation Research Part C: Emerging Technologies, vol. 85, pp. 591–608, 2017.
44. A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "DySAT: Deep neural representation learning on dynamic graphs via self-attention networks," in Proc. WSDM, 2020, pp. 519–527.

45. Ghimire S, Bhurtel N, Jha S, Ahmad S, Abdeljaber HA, Nazeer J. Minimizing energy consumption in fixed node networks using a novel neutrosophic model: comparative analysis with standard and existing algorithms. International Journal of Information Technology. 2025 Jan 9:1-5.
46. Haque MA, Sonal D, Ahmad S, Abdeljaber HA. Leveraging IoT for Wildlife Deterrence: Smart Solutions for Crop Protection in Modern Farming. IoT and Advanced Intelligence Computation for Smart Agriculture. 2026:126-39.
47. Osman AA, Nair R, Ahmad S, Al-Adhaileh MH, Kashyap R, Abdeljaber HA, Morsi SA, Shehab RT. Exploring Deep Learning Approaches for Multimodal Breast Cancer Dataset Classification and Detection. Data and Metadata. 2025;4:1136-.
48. Alzu'bi A, Albashayreh A, Abuarqoub A, Alfawair MA. Explainable AI-based DDoS attacks classification using deep transfer learning. Computers, Materials & Continua. 2024 Sep 12;80(3):3785-802.
49. Albashayreh A, Najadat H. Detecting the impact of covid-19 on social media using bert-based model. In2024 15th International Conference on Information and Communication Systems (ICICS) 2024 Aug 13 (pp. 1-6). IEEE.
50. ALDabbas A, Baniata LH, AlSaaidah BA, Mustafa Z, Alali M, Rateb R. Artificial intelligence-driven method for the discovery and prevention of distributed denial of service attacks. Int J Artif Intell ISSN. 2025;2252(8938):8938.
51. Bsoul Q, Zawaideh F, Alqadi BS, Almusfar LA, Khalaf OI, Alattas AS, Alali M, AbdElminaam DS. From user preferences to accurate predictions: enhancing movie recommendation systems with neural collaborative filtering and sentiment analysis. SN Computer Science. 2025 Mar 8;6(3):257.
52. Alshnaikat MS, Alsunbuli BN, Mahyuddin NM, Ismail W, Shamaileh AA, Barqawi MH, Farrah L, Ozturk BA. The communication MIMO system for performance of thermal noise, and nonlinearity losses of an RF transceiver phase noise an RF in 6G. Internet Technology Letters. 2025 Sep;8(5):e70087.
53. Alqsass M, Jebreel M, Dweiri M, Qabajeh M, Al-Hakim M, Al-Hamad AA, Almajali D. The Role of Artificial Intelligence Adoption to Enhance Financial Performance:(Case Study Based on Jordanian Traditional Banks). InInternational Conference on Technology and Innovation Management 2024 May 20 (pp. 555-561). Cham: Springer Nature Switzerland.
54. Jebril, J., Adnan, S., Jebreel, M., Al-Shanti, A., Al-Hamad, A.AS., Qubbaja, A. (2025). The Impact of Organizational Antecedents on the Business Intelligence Mediating Role of Organization's Culture in Insurance Companies. In: Alzoubi, H.M., Al-Gasaymeh, A.S., Vasudevan, S. (eds) Bridging Digital Innovation and Technology for Business Transformation—ICTIM. ICTIM 2024. Advances in Science, Technology & Innovation. Springer, Cham. https://doi.org/10.1007/978-3-031-90558-2_45
55. Beyari H, Hashem T. The role of artificial intelligence in personalizing social media marketing strategies for enhanced customer experience. Behavioral Sciences. 2025 May 19;15(5):700.
56. Alghizzawi M, Zahran I, Al Sokkar AA, Gasawneh JA, AlFraihat SF. Exploring the multifaceted impact of augmented reality applications across industries and consumer behavior. InKnowledge sharing and fostering collaborative business culture 2025 (pp. 363-376). IGI Global Scientific Publishing.
57. Masa'deh RE, Almajali DA, Alsmadi LA. Acceptance of Artificial Intelligence in a Jordanian Firm: An Overview.
58. Alghizzawi M, Jarrah MA, Al-Gasawneh J, Habes M, Elareshi M, Ziani A, Alsridi H. The Effects of Digitalization Features on Work Performance. InInternational Conference on Business and Technology 2024 Nov 21 (pp. 486-492). Cham: Springer Nature Switzerland.
59. Hanandeh A, ALFreijat SY, Qutieshat RJ, Alsha'ar HY, Kilani QA, Saleem Khasawneh MA. Implementing AI Accuracy, Learning Rate, Inference Time on enhancing Big Data Analysis and Strategic Plan. Data and Metadata. 2025;4:637.
60. Al Shurideh M, Alzoubi HM, Al Kurdi B, Hamadneh S, Ahmed G, Al-Sulaiti K, Bataineh AQ, Alquqa EK, Ozturk I. Consumer and economic influences on electric vehicle adoption: The mediating role of attitudes and the moderating effect of demographics. International Journal of Energy Economics and Policy. 2025;15.

61. Alhanatleh H, Khaddam A, Alfreihat H, Alghizzawi M, Hmeidan TM. Customers' Retention of Using Mobile Fintech Services: Telecommunication Companies in Jordan. InInternational Conference on Technology and Innovation Management 2024 May 20 (pp. 385-392). Cham: Springer Nature Switzerland.
62. Jawabreh O, Masa'deh RE, Al Fahmawee EA. Adoption of Artificial Intelligence Chatbots in Tourism in Jordan. InGenerative AI in Creative Industries 2025 Jul 5 (pp. 101-116). Cham: Springer Nature Switzerland.
63. Alkayed M, Almasalha F, Hijjawi M, Qutqut MH. Factors analysis affecting academic achievement of undergraduate student: A study on faculty of information technology students at applied science private university. In2023 International Conference on Business Analytics for Technology and Security (ICBATS) 2023 Mar 7 (pp. 1-12). IEEE.
64. Qusef A, Murad S, Alsalhi NR, Al Gharaibeh F. Leveraging artificial intelligence to identify students with learning challenges. International Journal of Learning, Teaching and Educational Research. 2025 May;24(5):623-43.
65. Abualhomos M, Shihadeh A, A Abubaker A, Al-Husban K, Fujita T, Alsaraireh AA, Shatnawi M, Al-Husban A. Unified framework for type-n extensions of fuzzy, neutrosophic, and plithogenic offsets: Definitions and interconnections. Journal of fuzzy extension and applications. 2025 Dec 1;6(4):689-726.
66. Shihadeh A, Abualhomos M, Abd El-latif AM, Alhusban A, Shaaban SM, Arslan M, Khattak AM. Key characteristics of quadri-partitioned neutrosophic riemann integrals and quadri-partitioned neutrosophic soft topological spaces. European Journal of Pure and Applied Mathematics. 2025 May 1;18(2):5839-.
67. Djedid Z, Batiha IM, Alsharif S, Abu-Ghurra S, Aljazzazi M, Jawdat J. Fixed point of hardy-rogers contraction mappings in non-solid cone\(\chi_b\)-metric space with applications. Adv. Fixed Point Theory. 2025 Jun 30;15:Article-ID.
68. AbuHijleh EA, Khalil SA, Abu-Ghurra S, Alafifi G. On the characterization of Harmonized fuzzy subgroups θ Open set. International Journal of Neutrosophic Science (IJNS). 2025 May 1;26(1).
69. Abubaker AA, Abualhomos M, Matarneh K, Al-Husban A. A Numerical Approach For The Algebra of Two-Fold 2× 2 Fuzzy Real Matrices and Anti Fuzzy Matrices. Neutrosophic Sets & Systems. 2025 Jan 1;75.
70. Aboshighiba H, Benariba A, Sbaa MR, Souad M, Sidamar L, Suleiman RK, Abu-Rayyan A, Meliani MH. Modeling of Textured Hydrostatic Thrust Bearings and Lubricating Films with Variable Thickness. Arabian Journal for Science and Engineering. 2025 Feb;50(4):2911-23.
71. Al-Billeh T, Al-Hammouri A, Al-Khalaileh L. Methods of teaching electronic administration legislation by using artificial intelligence techniques. International Journal of Electronic Security and Digital Forensics. 2025;17(3):351-62.
72. Tarawneh HA, Issa HA. Character Assassination via Information Technology Media: A New Crime. InInternational Conference on Technology and Innovation Management 2024 May 20 (pp. 203-208). Cham: Springer Nature Switzerland.
73. Alkhateeb, J., Ismail, M., Massadeh, F., & Almansour, H. (2025). Legal characterization of digital copyrights in non-fungible tokens (NFTs) form. In Advances in Science, Technology and Innovation (pp. 171–181). Springer. https://doi.org/10.1007/978-3-031-90558-2_22
74. Ahmad M, Haider A, Saed H. Assessing AI-driven dubbing websites: Reactions of Arabic native speakers to AI-dubbed English videos in Arabic. Humanities. 2025;6(1).
75. Darwish N, Haider A, Tannous B, Rumman RN, Alantari D, Saed H, Dagamseh M. A reception study of AI-translated idioms and proverbs between Arabic and English. Humanities. 2025;6(3).
76. Darwish N, Darwish T, Haider AS. Family relations in the age of social media: How Jordanians view the influence of Facebook and WhatsApp. Humanities. 2025;6(3).
77. Alomari MA, Khabour OF, Alzoubi KH, Aburub A. The impact of COVID-19 confinement on reading behavior. Clinical Practice and Epidemiology in Mental Health: CP & EMH. 2023 Jun 5;19:e174501792304260.

78. Alkasasbeh AM, Jarrah SS, Alhusamiah BK, Tarawneh FS. Factors influencing the utilization and adoption of electronic health records among nurses in Jordanian hospitals. Jordan Journal of Nursing Research. 2025;4(1):68-80.
79. Abu-Farha RK, Alzoubi KH, Al Safadi AA, Alsous MM, Nawasreh A, El-zubi MK, Al-Ashwal FY. Exploring Physicians' Willingness to Integrate Artificial Intelligence in Clinical Practice: Ethical and Practical Insights From a Jordanian Cross-Sectional Survey. Health Science Reports. 2026 Mar;9(3):e71994.
80. Ahmad N, Kaleem M, Elloumi M, Mushtaq MA, Fatnassi A, Fazil M, Bilal A, Darem AA. A Comprehensive Literature Review of AI-Driven Application Mapping and Scheduling Techniques for Network-on-Chip Systems. Computer Modeling in Engineering & Sciences. 2026;146(1).
81. Sajid M, Malik KR, Khan AH, Bilal A, Alqazzaz A, Darem AA. Advanced multilayer security framework: integrating AES and LSB for enhanced data protection: M. Sajid et al. The Journal of Supercomputing. 2025 Nov 29;81(17):1607.
82. Ajmal SU, Hussain M, Hashmi MU, Muslim FB, Bilal A. Hybrid Micro-Grid Operation during Islanded Mode using Sliding Mode Control. In2025 International Conference on Computer, Internet of Things and Smart City (CIoTSC) 2025 Nov 7 (pp. 1-14). IEEE.