ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 3 - 2026

Multi-scale Dual Transformer based Multi long-term Time Series Prediction and Dynamic Correction Method for Traffic Flow (MSD-Transformer)

Limi Chen[1,2,*], *Zhihao Jiang*[1,2] *and Jing Yang*[1]

[1] Hainan Vocational University of Science and Technology, Haikou 571126, China
[2] Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Malaysia

SUMMARY: *In response to the problem of congestion delay dependence in existing traffic flow prediction models, a dynamic correction method for multi-dimensional long-term time series prediction of traffic flow (MSD-Transformer) integrating multi-scale dual Transformers is proposed. Firstly, a multi-scale feature extraction network is constructed using weight sharing Swin Tiny, which hierarchically mines multi-dimensional traffic temporal features such as short-term fluctuations, daily cycles, and weekly cycles. A bidirectional feature pyramid BiFPN is introduced to build a bidirectional feature fusion pathway, and an adaptive weighting mechanism is used to achieve complementary interaction between high and low resolution features; Secondly, design a spatial residual attention module that combines layered residual attention scores to synchronize sequence self-attention and cross sequence cross attention modeling, accurately characterizing the differential congestion propagation delay of main and branch roads; Thirdly, based on the dual Transformer architecture, a multi-scale spatiotemporal encoding and decoding unit is built, and the model training optimization is completed by combining weighted cross entropy and focus loss fusion function. The experimental results show that compared with mainstream baseline models such as STGCN, DCRNN, and STTN, the proposed model significantly reduces MAE and RMSE indicators in the 15-60 minute full time prediction task; Experimental validation of multi-scale feature extraction network BiFPN, The three major modules of spatial residual attention all have irreplaceable gains. Research has shown that the proposed method can finely model traffic spatiotemporal delays and non-stationary sudden changes in flow, effectively alleviating the performance degradation of long-term time series prediction and providing precise support for dynamic flow regulation of urban road networks.*

KEYWORDS: *Traffic flow prediction; Multi-scale features; Dual Transformer; Bidirectional feature pyramid; Spatial residual attention; Spatiotemporal dependency modeling*

1 Introduction

With the acceleration of urbanization and the continuous increase in the number of motor vehicles, road congestion caused by the imbalance between supply and demand in urban transportation systems has become a bottleneck problem that restricts high-quality economic and social development [1]. As a core technical means of road network state perception and dynamic regulation, traffic flow prediction directly affects the effectiveness of control strategies such as signal timing optimization and guidance path planning, thereby determining the space

*chenlimi256@126.com
https://doi.org/10.65102/is20261301
Published: 8 July, 2026

for improving the operational efficiency of the transportation system. Traditional traffic flow prediction methods are mainly based on physical models and statistical methods, such as time series analysis and Kalman filtering [2]. They have a simple structure, high computational efficiency, and relatively low requirements for data volume; However, due to the strong nonlinearity and randomness of the transportation system, traditional methods often struggle to fully capture the inherent patterns in complex traffic dynamics, resulting in a bottleneck in prediction accuracy.

The development process of traffic flow prediction models can be divided into three main stages. Early research was mainly based on statistical methods. For example, Durán-López et al. [3] integrated ARIMA and Kalman filtering models to establish a hybrid prediction framework that includes Kalman filtering measurement equations and update equations, which to some extent improved the model's prediction accuracy. However, due to its model structure usually based on linear assumptions, it is difficult to fully characterize the nonlinear fluctuations, time-varying patterns, and complex periodic features commonly present in traffic flow data, which greatly limits its generalization ability and prediction accuracy. With the breakthrough of computing power, machine learning algorithms are gradually being applied in this field. For example, Mawarni & Hendrawan [4] introduces a random forest model to address the issue of high computational complexity in SVM. Experimental results have shown that this model outperforms SVM in terms of prediction accuracy, generalization ability, usability, and scalability. Although machine learning methods have overcome the limitations of linear models to some extent, their modeling process often relies on a large number of manually designed feature engineering, making it difficult to automatically extract deep spatiotemporal correlation features from traffic flow data. At the same time, there are performance bottlenecks when facing large-scale and complex structured data, which affects their promotion and application in practical scenarios. After entering the era of big data in the 21st century, deep learning algorithms have shown unique advantages in processing massive traffic data, mainly including architectures and variants such as recurrent neural networks (RNN), convolutional neural networks (CNN), graph neural networks (GCN), and Transformers. For example, Kwon et al. [5] developed a time information enhanced Transformer model, which combines long short-term memory networks and Transformer architecture to effectively extract spatiotemporal features of traffic flow and significantly reduce model complexity. Jinia et al. [6] proposes a CNN Attention Bi GRU hybrid model that integrates attention mechanisms to address the nonlinear and dynamic characteristics of traffic data. Reference [7] evaluated the real-time traffic situation by considering the spatiotemporal characteristics of urban transportation operation, and compared and analyzed the results between the Gated Recurrent Unit (GRU) and the traditional LSTM model, thereby improving the prediction accuracy of traffic operation situation. Although deep learning methods have significantly broken through the bottleneck of traditional models in feature extraction, there are still problems such as single model structure, insufficient generalization ability, lack of spatially dependent modeling, and high training costs.

This article proposes an MSD-Transformer method for multi-dimensional long-term time series prediction and dynamic correction of traffic flow, which integrates multi-scale dual Transformers. Based on weight sharing Swin Tiny, a multi-scale feature extraction network is constructed to capture short-term flow fluctuations, daily cycles, and weekly cycles in layers. Effective features are screened using CBAM channel spatial attention. Then, a bidirectional feature pyramid BiFPN is built to form an upper and lower bidirectional information flow path, and an adaptive weighting mechanism is used to achieve complementary fusion of high and low resolution features, solving the problem of traditional FPN unidirectional information flow being difficult to balance micro level mutations and macro level road network evolution. An innovative spatial residual attention module is designed to superimpose hierarchical residual

attention scores on the basis of standard multi head attention, synchronously completing sequence self-attention and cross sequence cross attention calculations, accurately distinguishing differentiated congestion propagation delays between main roads and branch roads; Finally, a dual Transformer spatiotemporal encoding and decoding unit is constructed, which integrates weighted cross entropy and focus loss composite functions to optimize model training. The experimental results show that the proposed model has significantly better MAE and RMSE indicators than mainstream baselines such as STGCN, DCRNN, and STTN in the 15–60-minute full time prediction task, which can effectively alleviate the long-term time series prediction accuracy decline and provide high-precision technical support for dynamic flow control of urban road networks.

2 Related work

The existing traffic flow prediction can be mainly divided into statistical models and machine learning based models, among which machine learning models can be further subdivided into recurrent neural network models, attention mechanism enhanced models, and dynamic graph neural network models.

2.1 Statistical based traffic flow prediction model

Statistical models, such as vector autoregression and its combination models, autoregressive moving average models, model temporal dependencies by linearly combining historical observations, but lack the ability to model nonlinear abrupt components of non-stationary traffic flow. Scholars have proposed a statistical nonlinear improvement model: the non-down sampling Shearlet transform extracts transient features through multi-scale decomposition, and the bilinear model introduces a product term to enhance nonlinear expression ability. However, such methods still rely on artificial feature design and are difficult to cope with high-dimensional spatiotemporal interaction scenarios.

2.2 Machine Learning based Traffic Flow Prediction Model

Machine learning models enhance their nonlinear modeling capabilities through kernel function mapping, such as Support Vector Regression (SVR) which utilizes the principle of structural risk minimization to optimize the prediction of hyperplanes. Leong [8] combines Principal Component Analysis (PCA) with multiple linear regression to achieve feature dimensionality reduction, but it is only applicable to univariate prediction tasks. Varshitha et al. [9] further combines grid search to optimize SVR hyperparameters, but its sliding time window mechanism ignores long-range dependency features. Traditional machine learning models are limited by shallow model structures and cannot effectively capture dynamic spatiotemporal coupling effects. Recent research on traffic flow prediction has focused on deep learning frameworks, with a focus on breaking through the dynamic modeling capabilities of spatiotemporal dependencies. Specifically, it can be divided into the following three categories:

(1) Recurrent neural network model. Long Short Term Memory (LSTM) is a special type of Recurrent Neural Network (RNN) used to solve the problem of gradient vanishing or exploding that traditional RNNs are prone to when processing long sequence data. Gated Recurrent Unit (GRU) is an improved recurrent neural network, similar to LSTM, designed to solve the problem of gradient vanishing or exploding in traditional RNNs when processing long sequence data, but with a more simplified structure The cascaded architecture of LSTM and GRU with Graph Convolutional Network (GCN) utilizes road network topology to construct a static adjacency matrix. However, the fixed aggregation weights of GCN are difficult to

characterize the time-varying characteristics of traffic conditions, such as sudden changes in morning and evening peak flow. Ali et al. [10] generates a dynamic adjacency matrix through meta learning, but does not explicitly model the delay propagation effect.

(2) Attention mechanism enhancement model. The Transformer based model utilizes self attention to capture global spatiotemporal dependencies. Kirubavathi et al. [11] proposed that causal sparse attention reduces computational complexity, but static sparse patterns are difficult to adapt to sudden traffic fluctuations. Thotla et al. [12] integrates an advanced architecture of multi-channel data input and spatiotemporal transformer to capture complex spatiotemporal features and interaction relationships in multidimensional data. However, its fixed window global attention mechanism leads to event response lag (average delay of 2.3 time steps).

(3) Dynamic graph neural network model. Alshehri et al. [13] implements large-scale road network modeling based on dynamic graph framework, but its spatial weight allocation does not consider the delayed propagation characteristics of traffic congestion; Hermosillo-Reynoso et al. [14] updates the graph structure through dynamic node embedding and differential equation updating. Izadkhah et al. [15] developed an adaptive delay graph convolutional layer that can learn the delay parameter Δt to adjust the message transmission speed, but adopts a uniform delay allocation strategy, ignoring the differential response of heterogeneous nodes to congestion propagation (such as the delay difference between main roads and branch roads can reach 8-15 minutes). Kumar et al. [16] proposed a dynamic Markov model to describe the evolution of time series graphs, but the Markov assumption limits its ability to model long delay chains.

2.3 Summary of Limitations of Existing Traffic Flow Prediction Models

There are three shortcomings in current research [17]: (1) coarse-grained modeling of delay dependence. The spatiotemporal dependency modeling commonly adopts synchronous aggregation strategy, ignoring the delay effect of traffic congestion propagation. (2) Lack of non-stationary modeling. The existing methods assume that traffic flow follows a steady or piecewise steady distribution, and lack robust adaptability to distribution shifts caused by unexpected events such as traffic accidents. (3) The inefficiency of dynamic topology updates. The update of graph structure relies on the similarity measurement of node attributes, and does not establish an explicit coupling relationship between propagation delay and network topology.

The above problems have led to a significant increase in prediction errors of existing models in complex urban road networks. In sudden congestion scenarios, mainstream models such as STG2Seq (Spatial-Temporal Graph to Sequence), Graph Wave Net, latest model Cycle LLH(Cycle Little Linear Head), CCHMM(Causal Conditional Hidden Markov Model) The mean absolute error (MAE) is significantly improved compared to stationary scenarios, further demonstrating the limitations of existing methods in dealing with non-stationarity and sudden traffic events.

3 Problem and Prediction Algorithm Framework

3.1 Problem Description

The traffic flow prediction task is a classic time series prediction task. In this study, historical time slices based on recent, daily, and weekly cycle lengths of T were used as input sequences [18]. Specifically, the three types of input sequences are fused into one input sequence consisting of $(a+b+g)$ subsequence segments. At this point, the problem of predicting traffic flow for the next α time steps is defined as follows:

$$\left[Y_{t+1},\cdots,Y_{t+q}\right]=f\left(\boldsymbol{X}_{\mathrm{w}},g,\boldsymbol{X}_{\mathrm{d}},b,\boldsymbol{X}_{\mathrm{r}},a\right) \tag{1}$$

where, a is the number of recent period segments; $\boldsymbol{X}_{\mathrm{r}}$ is the number of recent period segments; b is the number of daily period segments; $\boldsymbol{X}_{\mathrm{d}}$ is the number of daily period segments; $\boldsymbol{X}_{\mathrm{w}}$ is the number of weekly period segments; g is the number of weekly period segments.

The definitions of related concepts are as follows [19, 20].

Definition 1: Transportation network. Use undirected graph $G=\left(V,E,\boldsymbol{A}_{\mathrm{dis}},\boldsymbol{A}_{\mathrm{adp}}\right)$ to represent the transportation network, where V represents the set of nodes in the graph, with a total of N nodes; E represents the connectivity between nodes; $\boldsymbol{A}_{\mathrm{dis}}$ represents the predefined adjacency matrix of static spatial correlation between nodes; $\boldsymbol{A}_{\mathrm{adp}}$ represents the adaptive adjacency matrix of dynamic spatial correlation between nodes.

Definition 2: Traffic flow matrix. The traffic flow recorded by traffic network G at time t is the traffic flow matrix $\boldsymbol{X}_t=\left[\boldsymbol{x}_{t,1},\boldsymbol{x}_{t,2},\cdots,\boldsymbol{x}_{t,n}\right]$, where $\boldsymbol{x}_{t,v}$ represents the traffic flow of node v at time t.

Definition 3: Adaptive adjacency matrix. Infer the spatial dependency relationship between each pair of nodes through the dot product of node embedding vectors and Softmax normalization, and automatically update the adaptive adjacency matrix that reflects the dynamic spatial correlation between nodes through backpropagation during the training process. At the same time, utilizing the binary mask generated by Gumbel Sigmoid ensures the sparsity of the adaptive adjacency matrix.

Definition 4: Pre define adjacency matrix. Define the distance between nodes using the Manhattan distance function, and calculate the correlation between each pair of nodes through dynamic time warping (DTW) to construct a predefined matrix that reflects the static spatial correlation between nodes. To ensure comparability of matrix values, the matrix is normalized and a threshold of 0.6 is set. When the correlation between two nodes exceeds the threshold, the corresponding matrix element value is set to 1; Otherwise, set it to 0.

3.2 Basic Structure of Transformer

The basic structure of Transformer consists of four parts, namely feature embedding, multi head attention, position feedforward network, and position encoding. Self-attention maps the input traffic flow data $\boldsymbol{X}$, $\boldsymbol{Y}$ into three different sequences of vectors (Query: $\boldsymbol{Q}$, Key: $\boldsymbol{K}$, Value: $\boldsymbol{V}$), where the values of the vectors are the pixel values of the RGB channels, where n and d are the length and dimension of the input sequence, respectively. The calculation method for each vector is as follows:

$$\boldsymbol{K}=\boldsymbol{Y}\boldsymbol{W}^{K},\boldsymbol{Q}=\boldsymbol{X}\boldsymbol{W}^{Q},\boldsymbol{V}=\boldsymbol{Y}\boldsymbol{W}^{V} \tag{2}$$

where, $\boldsymbol{W}^{Q}$, $\boldsymbol{W}^{K}$, and $\boldsymbol{W}^{V}$ are linear weight matrices, respectively. $\boldsymbol{Q}$ is obtained by mapping $\boldsymbol{X}$, while $\boldsymbol{K}$ and $\boldsymbol{V}$ are obtained by mapping $\boldsymbol{Y}$.

When the input $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the same, it is called self-attention mechanism, and when the input $\boldsymbol{X}$ and $\boldsymbol{Y}$ are different, it is called cross attention mechanism. The self-attention mechanism is used in the encoding and decoding process of the Transformer module, while the cross-attention mechanism only plays a connecting role in the decoding process. The process of generating output features using vectors $\boldsymbol{K}$, $\boldsymbol{Q}$, and $\boldsymbol{V}$ is as follows:

$$\mathrm{Atten}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\mathrm{Softmax}\left(\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}/\sqrt{d_k}\right)\cdot\boldsymbol{V} \tag{3}$$

where, $\boldsymbol{K}^{\mathrm{T}}$ represents the transpose of $\boldsymbol{K}$, and the scaling factor $\sqrt{d_k}$ and Softmax function will pay attention to the weight normalization distribution.

To solve the problems of limited feature subspace size and poor global modeling ability of single attention, multi head attention mechanism MHA is applied in Transformer. MHA maps the input linear features into multiple feature subspaces and processes them in parallel into vectors through multiple attention heads. Finally, the resulting vectors are concatenated and output. This process can be expressed as:

$$\boldsymbol{O}_i = \mathrm{Atten}\left(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i\right) \tag{4}$$

$$\mathrm{MHA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{Concat}\left(\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots, \boldsymbol{O}_h\right) \cdot \boldsymbol{W}^o \tag{5}$$

where, h represents the number of attention heads; $\boldsymbol{W}^o$ represents the projection matrix of the output, and $\boldsymbol{Q}_i$ represents the output vector of each attention head. Multi head attention divides input features into h, and each attention head obtains d_{model}/h dimensional independent vectors in parallel to form head features.

The traffic flow data features generated by the multi head attention mechanism MHA are subjected to two consecutive linear transformations and ReLU activation functions to generate the traffic flow data features of the feedforward network. The expression is as follows:

$$\mathrm{FFN}(\boldsymbol{x}) = \mathrm{ReLU}\left(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1\right)\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{6}$$

where, $\mathrm{FFN}(\boldsymbol{x})$ is the traffic flow data feature output by the feedforward network.

When Transformer processes input traffic flow data, it does not retain the position information of the traffic flow data block sequence. In order to utilize the position information, an additional position vector called position encoding is often added to the input, which is expressed as:

$$\mathbf{PE}_t^{(i)} = \begin{cases} \sin\left(w_i t\right), & \text{if } k = 2i \\ \cos\left(w_i t\right), & \text{if } k = 2i+1 \end{cases} \tag{7}$$

$$w_i = \frac{1}{10000^{2i/d_{\mathrm{model}}}} \tag{8}$$

where, t represents the actual position of the element, $\mathbf{PE}_t$ is the position vector of the element, $\mathbf{PE}_t^{(i)}$ is the i-th element in the position vector, and d_{model} is the dimension of the element.

3.2 Traffic volume prediction algorithm framework

The MSD-Transformer network model structure proposed in this article is shown in Figure 1. The MSD-Transformer model mainly consists of a dual structure-based feature extraction network, three different scales and levels of multi-scale feature Transformer modules (ST, GT, and RT), and a shallow CNN structure prediction output module [21].

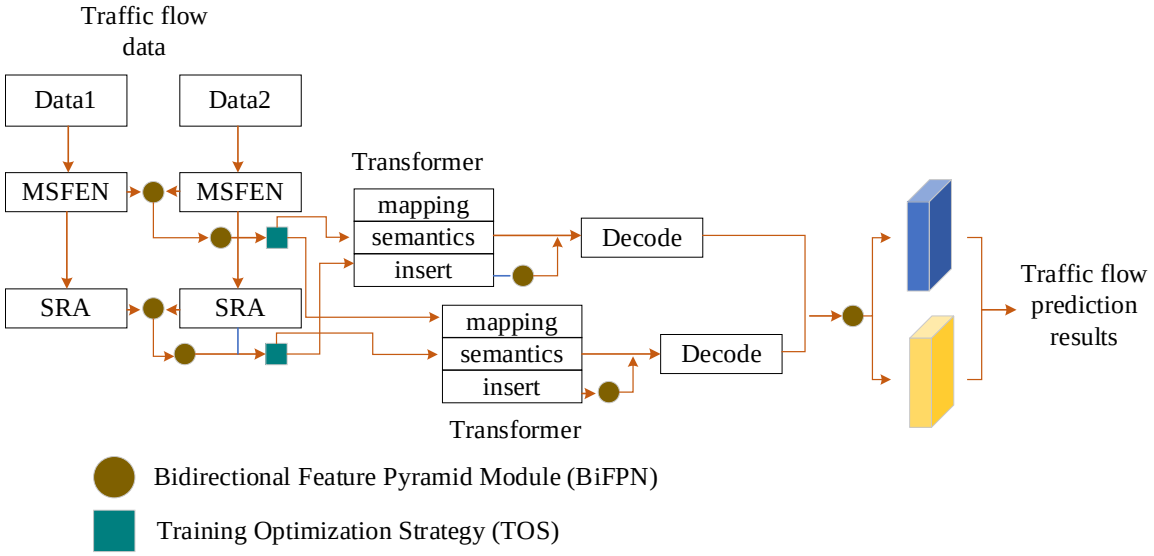

Figure 1: MSD-Transformer Model Structure Diagram

According to Figure 1, in the feature extraction stage of the twin structure, the first three layers of Res Net-18 are used to extract the original traffic flow data features, and the output traffic flow data feature sizes are $(1/4,1/8,1/16)$ of the input traffic flow data size ($H \times W$). The output traffic flow data features are concatenated with low-level dual temporal traffic flow data features through up sampling, and channel attention assigns channel weights to the concatenated features [22]. The fused traffic flow data features of different scales are fed into the multi-scale feature Transformer module for cross scale information exchange in the GT and RT modules, and cross spatial information exchange in the ST module. After ST decoding, refined dual temporal traffic flow data features are obtained. Finally, after subtracting the dual temporal features, a shallow CNN structure is used for discrimination to generate the final change detection result.

4 Description of Algorithm Improvement Module

4.1 Multi scale feature extraction network (MSFEN)

Given a traffic flow data frame, the inputs to the backbone are template traffic flow data $\boldsymbol{T}$, dynamic template traffic flow data $\boldsymbol{Z}$, and search traffic flow data $\boldsymbol{I}$. These traffic flow data are first input into the weight sharing Swin Tiny backbone network to generate feature embeddings.

The backbone network of the feature extraction module mainly consists of 4 layers of Swin Blocks and a multi-scale feature extraction module (MS-FEM). Input $3 \times H \times W$ traffic flow data into the Swin Tiny network, and the feature dimensions output by Swin Block 3 and Swin Block 4 are F3 and F4, respectively.

First, rearrange F4 and connect it with F3 along the channel dimension. When extracting features from template traffic flow data, spatial channel attention (CBAM) is added to obtain a high-quality target feature representation at the beginning of tracking, which facilitates the matching and recognition of targets in subsequent frames [23]. Subsequently, it is input into a Multilayer Perceptron (MLP) and the feature size is projected from 6C to D. C and D are set to 96 and 256, respectively. Finally, flatten the feature mapping and connect it along the spatial dimension using the following calculation formula:

$$\boldsymbol{F}_{\mathbf{M}} = \mathrm{MLP}\left(\mathrm{CBAM}_{T}\left[\boldsymbol{F}_{3} \oplus \mathrm{Rearrange}\left(\boldsymbol{F}_{4}\right)\right]\right) \tag{9}$$

$$\boldsymbol{X} = \mathrm{concat}\left(\boldsymbol{F}_{M}^{T}, \boldsymbol{F}_{M}^{Z}, \boldsymbol{F}_{M}^{I}\right) \tag{10}$$

where, The function of Rearrange is to improve the resolution of traffic flow data features while ensuring scale information; $\oplus$ represents connecting operations along the channel dimension; CBAM_{T} represents the addition of CBAM attention when extracting features from template traffic flow data; MLP stands for Multi-Layer Perceptron; concat represents concatenation; $\boldsymbol{F}_{M}^{T}$, $\boldsymbol{F}_{M}^{Z}$, and $\boldsymbol{F}_{M}^{I}$ respectively represent the output of template, dynamic template, and search traffic flow data connection; $\boldsymbol{X}$ represents the output of the multi-scale feature extraction network after feature connection, and N is the length of the feature sequence.

4.2 Bidirectional Feature Pyramid Module (BiFPN)

In the detection of changes in traffic flow data, high-resolution traffic flow data often contains change targets with significant scale differences, while the unidirectional information flow of traditional FPNs is difficult to simultaneously balance high-resolution details and low-resolution semantics. BiFPN is an improved structure of traditional FPN, which has efficient multi-scale feature fusion capability and is widely used in object detection tasks [24]. The core design is to introduce bidirectional information flow from top to bottom and bottom-up, combined with adaptive weighting mechanism, effectively enhancing the information interaction and semantic expression between features of different resolutions. For cross temporal multi-scale features in traffic flow data, BiFPN achieves complementarity through bidirectional paths: top-down paths transmit high-resolution details to capture subtle changes; Bottom-up path aggregation of low-resolution semantics to understand the logic of large-scale land cover changes. By combining dynamic weight allocation mechanism, key scale features can be strengthened for the non-uniform distribution characteristics of target scales in traffic flow data scenarios, improving the perception ability and positioning accuracy of multi-scale changing targets. Figure 2 shows the schematic structure of BiFPN, where the input traffic flow data features are bi directionally fused and weight optimized to generate enhanced features.

BiFPN achieves efficient information flow through both top-down and bottom-up bidirectional paths: high-resolution features convey detailed information downwards, while low resolution features convey semantic information upwards, thereby enhancing the perception ability of multi-scale targets. At each fusion node, the input traffic flow data features are first convolved with 1×1 to unify the number of channels, and then compressed through depthwise separable convolution to improve fusion efficiency and matching [25].

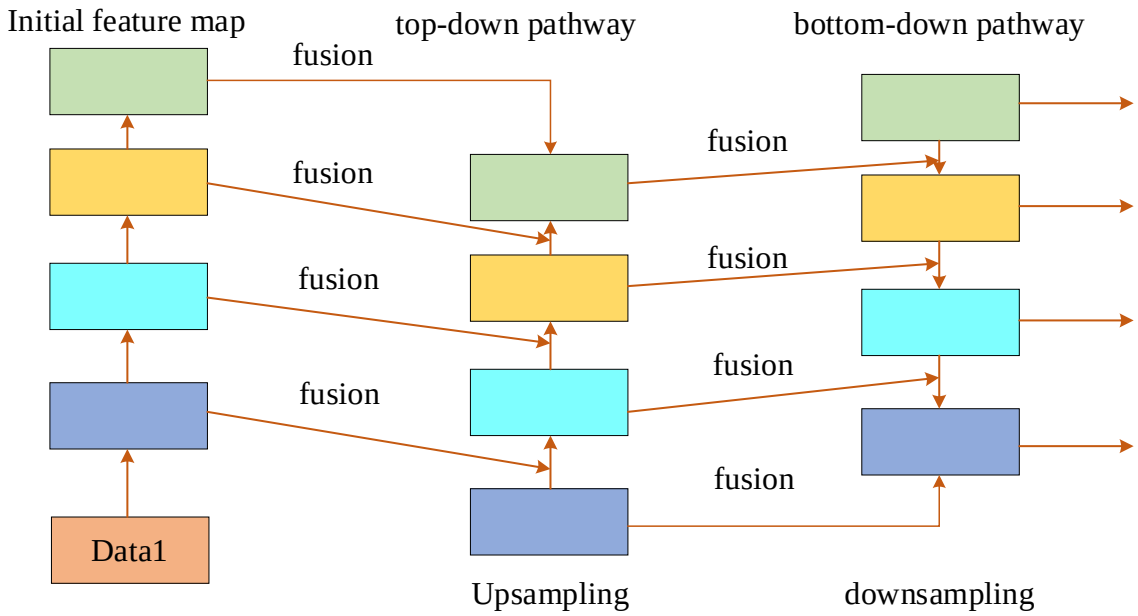

Figure 2: BiFPN module architecture

Meanwhile, BiFPN introduces an adaptive weighting mechanism to assign learnable weights to traffic flow data features from different sources, in order to enhance the contribution of key scale information. The weights are dynamically adjusted through training. The fusion formula is as follows:

$$F_{\text{fused},i} = \omega_1 \cdot P_i + \omega_2 \cdot P_{i+1} \tag{11}$$

where, ω_1 and ω_2 are the weights of each layer's traffic flow data features, while P_i and P_{i+1} are the low resolution and high-resolution traffic flow data features, respectively.

4.3 Spatial Residual Attention (SRA)

The Multi Head Self Attention (MHSA) mechanism is an important component of Transformer, and its main formation includes the following two steps [26]:

$$H_i = \text{Softmax}\left(\boldsymbol{X}\boldsymbol{W}_i^{Q} \cdot \left(\boldsymbol{X}\boldsymbol{W}_i^{K}\right)^{\text{T}} \Big/ \sqrt{d_k}\right)\boldsymbol{X}\boldsymbol{W}_i^{V} \tag{12}$$

$$\text{MultiHead}(q,k,v) = \text{concat}\left(H_1,\cdots,H_h\right) \tag{13}$$

where, $i = 1,\cdots,h$, h represents the number of heads; $\boldsymbol{W}^{Q}$, $\boldsymbol{W}^{K}$, $\boldsymbol{W}^{V}$ are linear weights that project the embedded $\boldsymbol{X}$ onto the query q key k and value v; d_k is the dimension of the key.

The proposed Spatial Residual Attention (SRA) first divides the input feature sequence $\boldsymbol{X}$ into target template feature $\boldsymbol{F}_T$ and search traffic flow data feature $\boldsymbol{F}_S$, and normalizes and reshapes the segmented feature sequence into a 2D feature map. Then, before performing planarization processing, separable deep convolutional projection layers will be executed on each feature map to achieve additional modeling of local spatial context for queries, keys, and values. After planarization processing, the queries, keys, and values required for attention operations are obtained through linear projection, where q_t, k_t, and v_t are projected from

the target sequence, and q_s, k_s, and v_s are projected from the search area. Finally, self-attention and cross attention are applied to the target sequence and search sequence, and the calculation formula is as follows:

$$\mathrm{Attn}_t = \mathrm{Softmax}\left(q_t \cdot k_t^{\mathrm{T}} / \sqrt{d}\right) v_t \tag{14}$$

$$\mathrm{Attn}_s = \mathrm{Softmax}\left(q_s \cdot \left(k_t \oplus k_s\right)^{\mathrm{T}} / \sqrt{d}\right)\left(v_t \oplus v_s\right) \tag{15}$$

where, $\oplus$ represents connection operation; Attn_t is the self-attention of template features; Attn_s is the cross attention between template and search traffic flow data.

Unlike standard multi head attention, this method performs dual attention operations. Firstly, for each sequence in the target template and search area, self-attention processing is performed separately to capture contextual information within each sequence. In addition, information exchange between the target template and the search area is achieved through cross attention, which enables the target template to better adapt to the target state in the current frame, enhance the dynamic capture ability of the target, and improve the robustness of tracking.

Unlike standard multi head attention, this method performs dual attention operations. Firstly, for each sequence in the target template and search area, self-attention processing is performed separately to capture contextual information within each sequence. In addition, information exchange between the target template and the search area is achieved through cross attention, which enables the target template to better adapt to the target state in the current frame, enhance the dynamic capture ability of the target, and improve the robustness of tracking.

In addition, when performing attention operations, the attention score of the previous layer before Softmax is added, that is, the residual attention score is added when calculating the current attention score, and then the weighted sum is calculated. The calculation formula is as follows:

$$\mathrm{ResidualAttn}(q,k,v,\mathrm{pre}) = \mathrm{Softmax}\left(q \cdot k^{\mathrm{T}} / \sqrt{d} + \mathrm{pre}\right) v \tag{16}$$

where, Pre represents the attention score of the previous layer. Residual Attn can transfer and accumulate attention information between different layers, enhance the feature representation of target objects, improve the robustness of target appearance changes and occlusion during tracking. Finally, the new attention score $\left(q \cdot k^{\mathrm{T}} / \sqrt{d} + \mathrm{pre}\right)$ will be passed on to the next layer.

4.4 Training Optimization Strategy (TOS)

To improve the generalization ability of the model and suppress overfitting, this paper introduces various data augmentation methods during the training phase, including random horizontal flipping, vertical flipping, 90 °rotation, and normalization operations. The above enhancement operation effectively expands the training sample space and improves the model's adaptability to changes in posture and perspective while maintaining semantic consistency of traffic flow data [27].

The model training adopts the AdamW (Adam with weight decay) optimizer, with an initial learning rate of 1×10^{-4} and a weight decay factor of 0.01. To achieve a smoother convergence process, a polynomial decay learning rate scheduling strategy is introduced, which takes the form of:

$$\alpha_t = \alpha_0 \cdot \left(1 - t/T\right)^{0.9} \tag{17}$$

where, α_0 is the initial learning rate, t represents the current training round, and T is the maximum number of training rounds.

In terms of loss function design, in order to balance the imbalance of sample categories and differences in sample difficulty, this paper combines weighted cross entropy loss and focal loss through weighted fusion [28]. The cross-entropy loss function is widely used in classification tasks, which can effectively drive the model to learn the correct discriminative boundary by measuring the difference between the predicted probability distribution and the true label. It is defined as:

$$L_{\mathrm{CE}}(p, y) = -(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)) \tag{18}$$

where, p is the positive probability predicted by the model, and y is the true label.

Focal loss introduces a focus factor γ to reduce the dominant role of easy to classify samples in the loss, thereby focusing on difficult to classify samples. The formula is:

$$L_{\mathrm{Focal}}\left(p_t\right) = -\alpha_t \cdot \left(1 - p_t\right)^{\gamma} \cdot \log\left(p_t\right) \tag{19}$$

where, α_t is a category weight used to balance the influence of positive and negative categories. γ is the focus factor (γ=2 in this article), and p_t is the current prediction probability.

Taking into account the advantages of both, the final loss function in this article is defined as:

$$L = \lambda \cdot L_{\mathrm{CE}} + (1 - \lambda) \cdot L_{\mathrm{Focal}} \tag{20}$$

where, $\lambda \in [0,1]$ is the loss weighting coefficient that controls the contribution ratio of the two loss functions. This combination strategy aims to improve the overall performance of the model.

5 Experimental analysis

5.1 Experimental dataset and experimental environment

To validate the method proposed in this study, experiments were conducted on the California datasets PEMS04 and PEMS08 in the United States. The PEMS dataset contains three features, namely traffic, speed, and occupancy. The relevant information of these datasets is shown in Table 1. Where, n_s is the number of sensors, t_o is the sampling interval. PEMS08 is traffic data collected from 168 testing points in San Bernardino in July August 2018; PEMS04 is traffic data collected from 302 monitoring points in the San Francisco Bay Area in January and February 2018. These data are organized into records every 5 minutes. In addition, standard normalization was used to process the data, and the training set, validation set, and test set were randomly divided in a 6:2:2 ratio. To evaluate the generalization performance of the model, multiple different validation sets are generated and the average value of the evaluation metrics is calculated on each validation set.

Table 1: Dataset Description

dataset	t_0/min	n_s	time range	timestamp
PEMS04	5	302	2018/01/01—2018/02/28	16987
PEMS08	5	168	2018/07/01—2018/08/31	17843

The hardware platform is as follows: 12th Gen Intel (R) Core (TM) i5-1240 @ 3.2GHz processor, NVIDIA Ge Force MX550 GPU. The software configuration is as follows: Windows 11 operating system, Anaconda 3 resource management, Pycharm IDE, Pytorch 2.4.0 deep learning framework, Python language development.

To verify the predictive performance of the proposed model, the following five deep learning-based traffic flow prediction models were selected as baseline models, and comparative experiments were conducted on the PEMS04 and PEMS08 datasets.

1) STGCN: By combining graph convolutional networks in the spatial domain and one-dimensional convolutional networks in the temporal domain, a complete convolutional module structure is constructed for extracting spatiotemporal features.

2) DCRNN: uses a diffusion convolutional network to learn spatial information of traffic flow data and employs a sequence-to-sequence model to capture time series.

3) DMSTGCN: An advanced spatiotemporal graph convolutional network model that focuses on dynamic and multi-dimensional spatiotemporal data processing.

4) Trendformer: follows the encoder decoder structure of Transformer, where the encoder structure encodes the input traffic flow to calculate the average trend, and the decoder outputs prediction results based on this trend.

5) STTN: Build a neural network architecture for spatial Transformer and temporal Transformer, dynamically capturing spatial and long-term dependencies between nodes in the transportation network through self-attention mechanism.

5.2 Result Analysis

The comparison results of different models using RMSE and MAE as evaluation metrics on the PEMS04 and PEMS08 datasets are shown in Table 2 and Table 3, respectively. Where, T_p is the actual prediction duration.

Table 2: Comparison results of RMSE and MAE for different models on the PEMS04 dataset

Model	T_p=60 min		T_p=45 min		T_p=30 min		T_p=15 min	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
STGCN	25.821	38.302	23.364	35.759	20.359	32.128	18.865	30.376
DCRNN	27.409	40.083	22.770	34.621	21.073	31.609	18.043	27.540
DMSTGCN	25.045	37.560	21.520	33.260	20.015	30.347	17.609	27.091
Trendformer	22.438	33.065	21.202	32.419	19.438	29.476	17.504	26.351
STTN	21.326	32.241	20.642	30.165	18.364	28.650	16.840	25.830
MSD-Transformer	17.653	27.784	16.830	26.340	16.307	26.418	15.210	24.651

Table 3: Comparison results of RMSE and MAE for different models on the PEMS08 dataset

Model	T_p=60 min		T_p=45 min		T_p=30 min		T_p=15 min	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
STGCN	25.875	18.650	28.341	20.416	31.753	23.027	33.470	24.853
DCRNN	23.348	16.821	26.470	19.257	30.376	22.653	31.803	24.016
DMSTGCN	23.008	16.469	25.731	18.731	28.239	21.074	30.631	23.569
Trendformer	22.602	16.210	25.032	18.471	27.721	20.732	29.458	22.958
STTN	21.653	15.843	24.360	18.042	26.265	19.812	28.018	21.854
MSD-Transformer	20.592	14.947	22.429	16.130	24.018	16.638	25.516	18.010

According to the experimental results in Table 2 and Table 3, the lower the MAE and RMSE values, the smaller the prediction error of the model. (1) On the PEMS04 dataset, the traditional graph convolution models STGCN and DCRNN have the highest overall error, with MAE exceeding 25 for 60 minutes, indicating that fixed graph topology is difficult to capture long-term nonlinear fluctuations in traffic; DMSTGCN introduces dynamic spatiotemporal convolution to slightly reduce errors, but still weaker than Transformer class models. Trendformer and STTN perform better than graph convolution baseline, while STTN relies on spatiotemporal dual attention mechanism to achieve optimal baseline results; MSD-Transformer, relying on multi-scale dual transformers and spatial residual attention, predicts a 17.2% decrease in MAE compared to STTN at 60 minutes and a 9.7% decrease in MAE compared to the optimal baseline STTN at 15 minutes, resulting in synchronous improvement in both long and short-term prediction accuracy. (2) The pattern of the PEMS08 dataset is consistent with that of PEMS04, and STTN still performs the best in the baseline, with a predicted MAE of 21.653 at 60 minutes; MSD-Transformer corresponds to a 4.9% and 5.7% decrease in MAE and RMSE compared to STTN, respectively. Comparing the results of the two datasets, it can be found that as the prediction time increases from 15 minutes to 60 minutes, the baseline model error increases significantly higher than MSD-Transformer, proving that the multi-scale feature fusion, bidirectional feature pyramid, and residual attention module proposed in this paper can effectively alleviate the problem of long-term prediction accuracy degradation.

5.3 Ablation Experiment

In order to investigate the roles of each component in the proposed model, five variants were designed, namely the algorithm for removing multi-scale feature extraction networks (w/o 4.1), the algorithm for removing bidirectional feature pyramid modules (BiFPN) (w/o 4.2), and the algorithm for removing spatial residual attention (w/o 4.3), and ablation experiments were conducted on the PEMSE04 and PEMS08 datasets. The ablation results using RMSE and MAE as evaluation metrics for different variants on the PEMS04 and PEMS08 datasets are shown in Table 4 and Table 5, respectively.

Table 4: Comparison results of RMSE and MAE for different variants on the PEMS04 dataset

Model	T_p=60 min		T_p=45 min		T_p=30 min		T_p=15 min	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o 4.1	25.650	35.053	22.038	35.132	20.237	30.074	17.015	25.682
w/o 4.2	18.602	29.316	18.107	28.339	20.164	29.563	18.674	27.540
w/o 4.3	24.758	34.219	18.336	27.658	17.630	27.208	25.983	32.134
MSD-Transformer	17.653	27.784	16.830	26.340	16.307	26.418	15.210	24.651

Table 5: Comparison results of RMSE and MAE for different variants on the PEMS08 dataset

Model	T_p=60 min		T_p=45 min		T_p=30 min		T_p=15 min	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o 4.1	25.215	33.659	25.315	28.196	29.313	24.528	27.674	21.843
w/o 4.2	21.092	30.825	24,769	26.037	28.064	24.157	28.479	23.076
w/o 4.3	22.347	26.549	29.834	27.683	27.853	26.260	30.465	28.609
MSD-Transformer	20.592	14.947	22.429	16.130	24.018	16.638	25.516	18.010

According to the experimental results in Table 4 and Table 5, it can be seen that the error indicators between the three ablation variants and the complete MSD-Transformer in the PEMS04 and PEMS08 datasets show that the higher the MAE and RMSE values, the greater the prediction bias. After removing any core module, the model accuracy shows a significant decline. (1) After removing the multi-scale feature extraction network (w/o 4.1), the variant MAE reached 25.650 in the 60 minute long-term prediction of PEMS04, an improvement of 45.3% compared to the complete model; The 60 minute predicted RMSE of the PEMS08 dataset increased to 33.659, far exceeding the original model's 14.947, indicating that the multi-scale module can extract short-term fluctuations, daily cycles, and weekly cycle traffic features in layers. Without it, the model cannot distinguish different scale temporal patterns, and the long-term prediction performance deteriorates the most severely. (2) When removing the BiFPN bidirectional feature pyramid (w/o 4.2), the predicted MAE of PEMS04 increased from 17.653 to 18.602 at 60 minutes, and the short-term predicted MAE of PEMS08 increased to 28.479 at 15 minutes. The results indicate that BiFPN bidirectional pathway can synchronously fuse low-level details and high-level semantics. Without it, the model is difficult to capture cross scale congestion propagation within the road network, and the prediction accuracy of short-term traffic mutation scenarios significantly decreases. (3) The performance degradation caused by removing spatial residual attention SRA (w/o 4.3) is the greatest. The 15-minute prediction of MAE by PEMS04 improved by 70.8% compared to the complete model; The RMSE of each duration of PEMS08 has exceeded 26. The experimental results show that standard multi head attention can only model spatiotemporal dependencies in a single layer, while SRA combined with residual attention scores accumulates spatial correlation information layer by layer, which can accurately characterize the differentiated congestion delay between main and branch roads. The above results indicate that the three improved modules work together to solve the shortcomings of existing models, such as delay dependence on coarse-grained, poor non-stationary robustness, and inefficient topology updates. The complete MSD-Transformer relies on multi module coupling to achieve optimal prediction accuracy in all time periods and multiple offline networks.

6 Conclusion

This article proposes the MSD-Transformer, a dynamic correction model for multi-dimensional long-term traffic flow prediction that integrates multi-scale dual Transformers. The model is compared, validated, and analyzed on a publicly available road network dataset. The main research work can be summarized into three aspects: (1) constructing a multi-scale feature extraction network based on weight sharing to capture three types of differentiated traffic time series features: short-term instantaneous fluctuations, daily cycles, and weekly cycles. By using CBAM channel spatial attention to enhance effective feature weights and relying on MLP to unify feature dimensions, the problem of traditional single scale networks being unable to distinguish multi period traffic patterns is solved, providing complete multi-source feature

inputs for long-term time series prediction. (2) Introducing a bidirectional feature pyramid BiFPN to build an up and down bidirectional feature fusion pathway, combined with adaptive learnable weights to achieve high and low resolution feature complementarity, to compensate for the shortcomings of traditional unidirectional FPN, which can only transmit information unidirectionally and is difficult to synchronously balance micro traffic changes and macro road network evolution. (3) Design a spatial residual attention module SRA, which introduces a hierarchical residual attention score accumulation mechanism on the basis of standard multi head attention, synchronously executes sequence self-attention and cross sequence cross attention, and accurately models the differentiated congestion delay of main and branch roads; Building a multi-scale spatiotemporal encoding and decoding unit based on a dual Transformer architecture, while integrating weighted cross entropy loss and focus loss to construct a composite loss function, to alleviate the problems of imbalanced samples and insufficient learning of difficult to distinguish traffic samples. The overall experiment proves that the proposed method can finely model the spatiotemporal delay relationship of traffic, effectively suppress the decline of long-term time series prediction accuracy, and provide high-precision flow data support for urban signal timing, path guidance, and dynamic control of road networks.

Subsequent research can be deepened from three aspects: (1) The existing adaptive adjacency matrix of the model only relies on node embedding for static updates, without real-time fusion of dynamic event information such as road control and accidents. In the future, a graph differentiation real-time update mechanism can be introduced to construct an event driven dynamic topology graph, achieving dynamic adjustment of road network topology with traffic events. (2) The current model only utilizes three types of static sensor data: traffic, speed, and occupancy rate, without integrating external influencing factors such as weather, holidays, and large-scale events. The next step is to construct a multimodal input fusion module to embed external heterogeneous data into a multi-scale Transformer encoding layer, further improving the stability of extreme scene prediction. (3) The current experiment is based on an offline static dataset, without deploying online real-time prediction scenarios. In the future, the model structure can be lightweight, and the computational cost can be reduced through model pruning and quantization compression. An end-to-end real-time traffic prediction system can be built to implement real urban road network online regulation scenarios. (4) Further expand the applicability of the model and transfer the algorithm to multi city travel time series prediction tasks such as shared bicycles and bus passenger flow, to verify the model's general generalization ability.

Funding

This research was supported by the Youth Program of the Natural Science Foundation of Hainan Province, China, grant number 625QN349.

References

[1] Omar M, Yakub F, Wijaya A A, et al. Evaluation of encoder-only transformer for multi-step traffic flow prediction[J]. IEEE Access, 2025, 13: 106349-106368.

[2] Sattarzadeh A R, Pathirana P N. Traffic state estimation with spatio-temporal autoencoding transformer (STAT model)[J]. IEEE Access, 2025, 13: 87048-87067.

[3] Durán-López A, Bolaños-Martinez D, De S, et al. GCT: a Granger-causal transformer for multivariate traffic analysis in smart villages[J]. ACM Transactions on Intelligent

Systems and Technology, 2026, 17(2): 1-25.

[4] Mawarni E, Hendrawan A. Hybrid deep learning for spatiotemporal traffic forecasting: Integrating LSTM, transformer, and graph convolutional networks on the METR-LA dataset[J]. International Journal of Data Science, 2025, 6(2): 103-112.

[5] Kwon Y, Ahn S, Cho M, et al. Exploring the unseen: A transformer-based unknown traffic detection scheme with contextual feature representation[J]. Computer Networks, 2025, 265: 111286.

[6] Jinia S N, Azad S B, Akter R, et al. GTN-GCN: Real-Time Traffic Forecasting Using Graph Convolutional Network and Transformer[J]. Applied Computational Intelligence and Soft Computing, 2025, 2025(1): 5572638.

[7] Venkatasubramanian S, Ch S K B P K, Babu B P. Overcoming dataset imbalances and computational challenges in iot intrusion detection: A smote-enhanced transformer-based model[C]//2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT). IEEE, 2025: 1195-1204.

[8] Leong W Y. Generative AI-Powered Traffic and Mobility Solutions for Next-Generation Smart Cities[C]//2025 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2025: 679-680.

[9] Varshitha J S, Bhargav M, Kushal M, et al. Intrusion Detection System Based on Multi-Level Feature Extraction Using FCN Transformer Graph SAGE, GAT and Inductive Network[C]//2026 Second International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI). IEEE, 2026: 1-8.

[10] Ali A, Ali R, Asad M, et al. Exploiting attention-driven weather-aware multimodal spatio-temporal fusion for urban traffic flow prediction[J]. Future Generation Computer Systems, 2026: 108559.

[11] Kirubavathi G, Sumathi I R, Mahalakshmi J, et al. Detection and mitigation of TCP-based DDoS attacks in cloud environments using a self-attention and intersample attention transformer model: KG et al[J]. The Journal of Supercomputing, 2025, 81(3): 474.

[12] Thotla S B, Vyshnavi S, Anusha P, et al. Traffic congestion prediction using real time data by using deep learning techniques[J]. International Journal of Engineering & Extended Technologies Research (IJEETR), 2026, 8(2): 489-494.

[13] Alshehri M, Wu T, Almujally N A, et al. UAV-based intelligent traffic surveillance using recurrent neural networks and Swin transformer for dynamic environments[J]. Frontiers in Neurorobotics, 2025, 19: 1681341.

[14] Hermosillo-Reynoso F, López-Pimentel J C, Ruiz-Ibarra E, et al. A Transformer-Based Multi-Task Learning Model for Vehicle Traffic Surveillance[J]. Mathematics, 2025, 13(23): 3832.

[15] Izadkhah S, Rekabdar B, Wagner A, et al. A time series transformer attention model for enhancing bicyclist volume estimation using data fusion and feature selection techniques[C]//2025 19th International Conference on Semantic Computing (ICSC).

IEEE, 2025: 60-67.

[16] Kumar S S, Mishra A, Mohmedmhdi H, et al. Transformer-Based 3D Point Cloud Object Recognition for Autonomous Vehicles[C]//2025 3rd International Conference on Cyber Resilience (ICCR). IEEE, 2025: 1-7.

[17] Borra C R, Rayala R V, Pareek P K, et al. Advancing IoT security with temporal-based Swin transformer and LSTM: A hybrid model for balanced and accurate intrusion detection[C]//2025 International Conference on Intelligent and Cloud Computing (ICoICC). IEEE, 2025: 1-7.

[18] Pérez B, Resino M, Seco T, et al. Innovative approaches to traffic anomaly detection and classification using AI[J]. Applied Sciences, 2025, 15(10): 5520.

[19] Antari A, Abo-Aisheh Y, Shamasneh J, et al. Network traffic classification using machine learning, transformer, and large language models[C]//2025 IEEE 4th International Conference on Computing and Machine Intelligence (ICMI). IEEE, 2025: 1-5.

[20] Pal S, Sarkar M, Nagaraj S. A Small, Fast, Quantized Transformer Based Neural Network for Bitrate Prediction[C]//2025 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). IEEE, 2025: 1-6.

[21] Ali M, Saleem Y, Hina S, et al. DDoSViT: IoT DDoS attack detection for fortifying firmware Over-The-Air (OTA) updates using vision transformer[J]. Internet of Things, 2025, 30: 101527.

[22] Dehkordi S B, Nasri S, Dami S. Unveiling anomalies: transformative insights from transformer-based autoencoder models[J]. International Journal of Computers and Applications, 2025, 47(1): 29-44.

[23] Salehiyan A, Moghaddam P S, Kaveh M. An optimized Transformer–GAN–AE for intrusion detection in edge and IIoT systems: experimental insights from WUSTL-IIoT-2021, edgeIIoTset, and TON_IoT datasets[J]. Future Internet, 2025, 17(7): 279.

[24] Albalooshi F A. Advancing Urban Planning with Deep Learning: Intelligent Traffic Flow Prediction and Optimization for Smart Cities[J]. Future Transportation, 2025, 5(4): 133.

[25] Nelson S C, Jayakumari R B, Sheeba S L. Enhanced Heterogeneous Vehicular Networks With Intelligent Congestion Avoidance Mechanism via Regularized Q-Value-Based Graph Generalized Neural Network Transformer[J]. International Journal of Communication Systems, 2025, 38(12): e70151.

[26] AboulEla S, Kashef R. Enhancing iot intrusion detection with transformer-based network traffic classification[C]//2025 IEEE International systems Conference (SysCon). IEEE, 2025: 1-8.

[27] Ankireddy P, Gopalakrishnan S, Reddy V L. Gradient-enhanced focal-pooling vision transformer with adaptive tuning for robust and accurate vehicle detection in smart environments[J]. Iran Journal of Computer Science, 2025, 8(4): 1597-1614.

[28] Anaedevha R N, Trofimov A G. Stochastic Multimodal Transformer with Uncertainty

Quantification for Robust Network Intrusion Detection[C]//International Conference on Neuroinformatics. Cham: Springer Nature Switzerland, 2025: 428-447.