DGA-NCDE: Dual-Graph Attention Neural Controlled Differential Equation for Accurate Urban Rail Passenger Flow Prediction

Lin Bai, Hong Dai*, Shuang Wang

***Abstract*—Urban rail transit systems constitute a crucial component of modern transportation engineering, where accurate predictions of station-level passenger flow are essential for optimizing the efficiency of rail transit operations. Traditional spatio-temporal models, such as graph-based models and recurrent neural networks, typically use static approaches. These methods often require high-dimensional parameter spaces to capture the complex dynamics of urban rail systems, resulting in high computational costs and low model efficiency. Moreover, it is difficult for traditional graph learning models to dynamically adjust their focus when capturing changes and subtle patterns, finally failing to fully utilize heterogeneous information between nodes. As a result, these limitations prevent models from adapting to the evolving characteristics of urban rail passenger flows and identifying fine-grained patterns with precision.**

To address these challenges, this study presents the Dual-Graph Attention-Neural Controlled Differential Equation (DGA-NCDE) model, which combines local geographical associations and global semantic associations through a novel Dual-Graph Attention (DGA) Module. The DGA Module is seamlessly integrated into the Neural Controlled Differential Equation (NCDE) framework, enabling the model to dynamically adapt to urban rail passenger flow patterns.

To validate the effectiveness of the DGA-NCDE model, comprehensive experiments were conducted on two large-scale public datasets, HZMetro and SHMetro. When compared to the best baseline models, DGA-NCDE reduces MAPE by 3.44% on HZMetro and by 2.27% on SHMetro, while using just 1.12% of the parameters. In addition, the DGA-NCDE model significantly cuts the training time by approximately 51.31%, and the inference time by about 87.23%, demonstrating the significant improvement in both accuracy and efficiency of the DGA-NCDE model.

***Index Terms*—spatio-temporal forecasting, graph attention network, NCDE, dynamic system**

I. Introduction

In recent years, the accelerated pace of urbanization and the expanding population size underscore the critical importance of efficiently managing urban transportation systems. In this context, urban rail transit emerges as a pivotal, swift, and convenient mode of public transportation. To address the challenges posed by passenger flow congestion, accurate passenger flow forecasting has become a paramount priority.

The advancement of deep learning technology revolutionized the field of passenger flow prediction.. Researchers have increasingly turned to innovative methods, with Convolutional Neural Networks (CNN) and Graph Convolutional Neural Networks (GCN) emerging as popular choices for extracting temporal and spatial features from rail transit stations, respectively. CNN, a well-established deep learning model, boasts a range of variants that have been effectively utilized in diverse feature extraction tasks [1-4]. GCN, on the other hand, stands out for its ability to handle irregular graph structures, thus making it suitable for capturing complex correlations within passenger flow data at each station. One widely used association graph in rail transit stations is the geographic adjacency graph, employed as input data to facilitate the learning of node and edge representations through GCN. Fig. 1 provides a schematic diagram illustrat-ing the association among urban rail transit stations, wherein (A) represents the geographical topology of the urban rail transit network. and (B) depicts the local geographic association between sites in the traditional manner and the global semantic association targeted for extraction in this study.

As illustrated in Fig. 1(A), some stations are shown to be directly connected by traffic lines, indicating their physical proximity on a traditional association chart with corresponding edges represented by straight solid lines in Fig. 1(B). This association represents a low-order geographical connection between stations. In addition to geographical proximity, some stations also exhibit similarities in their functional semantics, resulting in a comparable flow of passengers between them. However, stations with analogous functions may lack a direct traffic line between them, causing a discrepancy in the geographical adjacency relationship, as illustrated in Fig. 1(B). Such a discrepancy implies that the geographical connection in real space might not fully reflect their dynamic similarity.

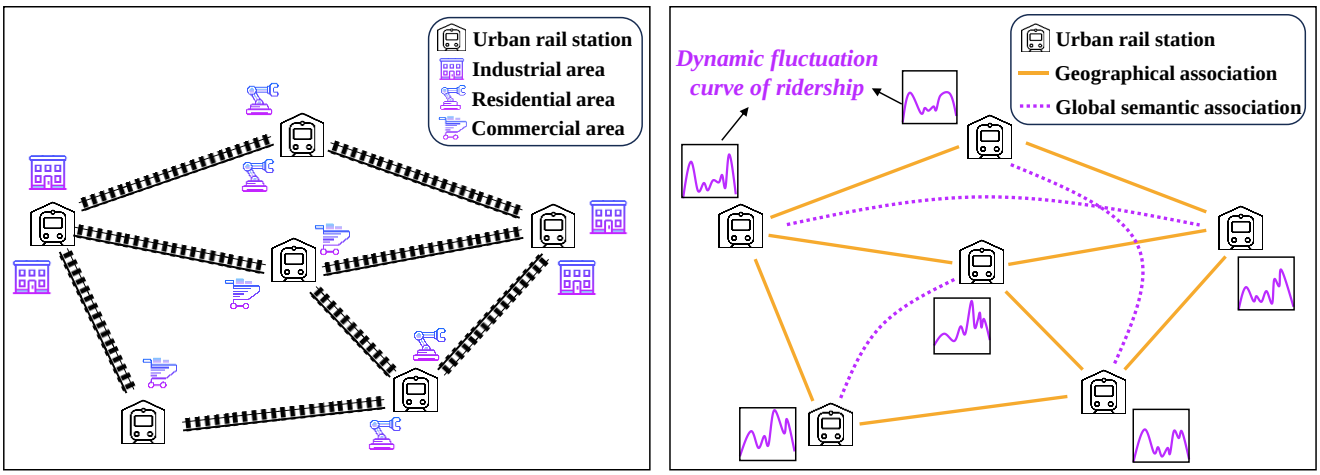
To address this issue, the incorporation of global semantic adjacency becomes crucial for capturing high-order passenger flow correlations among urban rail transit stations in the spatio-temporal prediction model developed in this paper, as indicated by the dotted line in Fig. 1(B).

Manuscript received January 17, 2024, revised May 29, 2024.

Lin Bai is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051, China. (e-mail: 1294330160@qq.com).

Hong Dai* is a professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China. (corresponding author to provide phone: +086-186-4226-8599; fax:0412-5929818; e-mail: dear_red9@163.com).

Shuang Wang is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051, China. (e-mail: 1657669526@qq.com).



(A) Low-order geographic connections **(B) High-order semanticconnections**

Fig. 1. Schematic diagram of the association of urban rail transit stations

GCNs excel in capturing intricate patterns and dependencies within static data, showcasing the capability to discern relationships among nodes in a graph, allowing for a more comprehensive comprehension of the underlying structure for data representation. As a result, GCNs are well-suited for tasks like node classification, recommendation systems [5], and text processing.

Nevertheless, challenges arise when applying GCNs to dynamic systems or datasets that undergo constant changes. Traditional GCNs give equal weight to all connections without considering the necessity of selectively focusing on pertinent information. This limitation becomes evident in tasks where certain nodes or connections hold more significance in predicting passenger flow, such as during peak hours or at major transfer stations. Consequently, a critical aspect of this study lies in improving the flexibility of the graph learning model to cope with the evolving dynamics and variations in urban rail passenger flow patterns.

On the flip side, the passenger flow within rail transit systems is a dynamic and evolving system influenced by numerous factors. Given the dynamic nature of urban rail transit passenger flow, Recurrent Neural Networks (RNN) [6] find widespread application due to their proficiency in modeling sequential data. However, the traditional RNN model face challenges in gradient computation and long-term memory when handling long-term spatio-temporal dependencies. In contrast, the Transformer model, known for its robust contextual feature extraction capabilities [7], struggle to seamlessly integrate complex spatial information. Additionally, the number of parameters in RNN-like models is highly influenced by the complexity of the data being predicted, resulting in significant parameter growth and increased computational costs, as well as challenging and costly model training. These issues are especially pronounced in complex, dynamic environments such as real-world traffic systems. Due to inefficient parameter usage, these models have limited practical applications. Hence, researchers are driven to explore more lightweight and efficient models to meet the growing demands for computational cost and efficiency.

In the pursuit of addressing we address the challenges posed by dynamic urban rail transit passenger flow prediction by exploring the use of Neural Controlled Differential Equation (NCDE) [8]. The NCDE presents a framework for modeling dynamic systems by directly incorporating control inputs, offering the advantages of continuous-time modeling and enabling the model to capture system evolution in a more natural and adaptive manner. To model the continuity of underlying dynamics in passenger flow and boost the model's adaptability to dynamic changes, this study employs the Graph Attention Network (GAT) [9] to capture more representative spatio-temporal dependencies. Moreover, GAT is integrated it into the NCDE framework for continuous dynamic system modeling. By combining GAT's spatial feature-capturing capability with the continuous dynamic perspective provided by NCDEs, the resulting model can better understand and predict the spatio-temporal dynamic changes of rail transit passenger flow. This innovative approach seamlessly integrates spatial and temporal dependencies, enabling the model to more precisely capture the complex dynamic features inherent in rail transit passenger flow. The main contributions of this paper include:

1) Introducing a Novel Method for Constructing a Global Semantic Adjacency Matrix: In response to the limited expressive capacity of the geographic adjacency graph, this paper puts forward a method for constructing a global semantic adjacency matrix. By leveraging global sequence similarity, this approach aims to depict the high-order associations of passenger flow among various stations effectively.
2) Proposing a Dual Graph Attention (DGA) Module to Enhance Graph Learning Model Adaptability: To enhance the adaptability of the graph learning model to the irregularities of urban rail passenger flow, this study proposes a DGA Module. In this network, two types of correlation graphs are successively input, allowing for the gradual capture of more detailed spatio-temporal correlation features.
3) Developing a Dual-Graph Attention-Based Neural Controlled Differential Equation Model (DGA-NCDE) for Dynamic Modeling: This paper presents an innovative modeling framework for understanding the underlying dynamics within passenger flow. The integration of the Dual Graph Attention Module into the

NCDE model treats passenger flow evolution as a sophisticated dynamic system. The rate of change in the cubic spline-interpolated passenger flow series serves as the control signal for the NCDE while the dual-graph attention network captures system dynamics. This integration significantly reduces model parameters and achieves higher prediction accuracy compared to larger baseline models. To the best knowledge of the author, this research represents the first successful application of NCDE in urban rail passenger flow forecasting and the first integration of graph attention networks with NCDE.

II. Related Work

A. Single Time Series Feature-Based Passenger Flow Prediction

In the domain of passenger flow prediction, leveraging single time series features is a classic approach for both modeling and forecasting. This method entails a detailed analysis of time series data, focusing on identifying trends, seasonality, and periodicity to thoroughly understand dynamic shifts in passenger flow. Among these techniques, the ARIMA model is particularly prominent, renowned for its effectiveness in modeling stationary time series. Li et al. in 2018 demonstrated the robustness of the ARIMA model in accurately fitting passenger flow data at Sanya Airport, underscoring its practical utility [10]. Additionally, Ding et al. improved traditional forecasts by combining Generalized Autoregressive Conditional Heteroskedasticity (GARCH) with ARIMA, enhancing the accuracy of rail transit passenger flow predictions [11].

As data volumes grew and accuracy demands intensified, researchers recognized the limitations of ARIMA in processing nonlinear, non-stationary, and high-dimensional time series data. This led to the widespread adoption of neural networks to capture temporal features, with techniques such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) gaining prominence for their ability to handle nonlinearities and long-term dependencies effectively. In 2019, Mou et al. introduced the Time-Domain Information Enhanced LSTM (T-LSTM), a novel method to predict single-section traffic flow. This approach proved capable of restoring original data features, significantly improving accuracy in traffic flow predictions [12]. That same year, Guo et al. developed the SVR-LSTM model, which combines Support Vector Regression(SVR) and LSTM neural networks to more effectively detect abnormal fluctuations in passenger flow [13].

To accurately capture the temporal dynamics of traffic flow, Zhao et al. employed the Temporal Convolutional Network (TCN) model. This model leverages a one-dimensional convolutional neural network architecture to process long sequence dependencies efficiently [14]. In 2020, Sha and others utilized the GRU network model for rolling predictions ranging from 15 minutes to 6 hours, greatly enhancing safety warnings and passenger flow management [15]. In 2023, VN. Katambire and colleagues innovated by combining the LSTM and ARIMA models to enhance traffic flow predictions. This integration significantly boosted the reliability of traffic flow scheduling [16].

These models excel in handling time series segments but falter with the spatial and temporal correlations in high-dimensional scenarios like site-level passenger flow. This limitation underscores the need for advanced methods to model complex spatio-temporal relationships effectively.

B. Spatio-temporal Correlation-Based Passenger Flow Prediction

Building on the limitations of traditional time-focused models, the evolution of traffic passenger flow is recognized as a complex spatio-temporal dynamic process influenced by both time and spatial location. Recent developments emphasize integrating these spatio-temporal characteristics into modeling and prediction strategies more effectively.

To effectively model spatio-temporal relationships, Zhang et al. in 2020 introduced a combined approach using residual networks and Graph Convolutional Networks (GCNs), integrated with LSTM, which accurately predicted short-term passenger flow on an urban rail transit network scale [17]. That same year, Peng et al. developed a dynamic graph neural network framework, outperforming traditional methods in urban traffic passenger flow forecasting [18]. In 2021, Wang et al. introduced a dynamic spatio-temporal hypergraph neural network, achieving even higher prediction accuracy [19]. Additionally, Yang et al. proposed an improved Spatio-temporal Long Short-Term Memory model (Sp-LSTM) to enhance vehicle scheduling, particularly in large transportation hubs [20].

Building upon these studies, Liu et al. in 2021 developed an end-to-end prediction model for subway ridership, named the Physical-Virtual Correlation Graph Network (PVCGN) [21]. PVCGN's graph convolution operation separates the physical-virtual graph into physical connection, similarity, and correlation graphs. By combining graph convolution and GRU, PVCGN achieves superior prediction results, surpassing other models. In 2023, Wang et al. introduced a traffic prediction model without a predefined structure. They utilized a temporal branch network and optimized graph neural Ordinary Differential Equations (ODE) to extract spatio-temporal dependencies, enhancing the model's effectiveness and stability [22].

However, traditional graph learning methods encounter certain challenges in predicting highly complex site-level passenger flows. Uniform weighting of all neighboring nodes can lead to imbalances, overemphasizing less critical nodes while neglecting more important ones. This imbalance hampers the capture of spatio-temporal relationships in passenger flow data. Additionally, static modeling methods lack the flexibility and adaptability needed to handle the dynamic nature of passenger flow. To address these challenges and improve prediction accuracy, more refined and adaptable modeling approaches that can dynamically adjust to evolving passenger flow patterns are required.

III. Preliminaries

A. Problem Formulation

The challenge of forecasting passenger traffic in rail transit is indeed intricate, yet at its core, it is essentially a spatio-temporal sequence prediction task. Let the passenger flow signal at station s at time t be denoted as x_t^s, and the entire sequence of signals for all stations is defined as

$X_t = [x_t^1, x_t^2, ..., x_t^N] \in \mathbb{R}^{N\times 2}$, where N is the total number of rail transit stations to be predicted. We can use the historical passenger flow at each station as the known temporal sequence signal along the time dimension and the adjacency graph G among the rail transit stations as the spatial correlation feature. Therefore, this task involves fitting a mapping function f that combines both temporal and spatial data and features. This mapping function is then applied to predict the future passenger flow at each station:

$$\left[X_{t-T+1}, X_{t-T+2}, ..., X_t, G\right] \xrightarrow{f} \left[X_{t+1}, X_{t+2}, ..., X_{t+T'}\right] \quad (1)$$

In the equation above, T represents the number of historical flow sequence steps, and T' represents the number of steps for predicting the passenger flow sequence. It is worth mentioning that in practical passenger flow prediction tasks, the passenger flow at rail transit stations is often divided into two types: entry flow and exit flow. Both types of passenger flows can be separately represented by the above equation to depict their prediction processes.

B. Neural controlled differential equations

Neural Ordinary Differential Equations (Neural ODEs) in graph neural networks offer a continuous perspective on discrete neural networks. It introduces and integrates an additional time-like dimension for residual networks. This dimension is artificially generated and is a necessary condition for the continuousization of discrete residual networks. The classical form of Neural ODEs is given by the following equation:

$$y(t_0) = y_0, y(T) = y(t_0) + \int_{t_0}^{T} f_\theta(t, y(t))\mathrm{d}t \quad (2)$$

Where $y(t_0)$ is the initial state, $t \in (t_0, T]$ represents a continuous time-like dimension, the system dynamically evolves from time t_0 to time T, and $f : \mathbb{R}\times\mathbb{R}^{u\times v\times ...\times z} \to \mathbb{R}^{u\times v\times ...\times z}$ can be any standard neural network architecture. θ represents the network parameters. However, in time series or spatio-temporal sequence prediction tasks, our historical data already inherently exhibits a temporal sequential structure. This implies that the artificially generated time-like dimension t may not be suitable for its inherent temporal modeling.

Given a series of time series signals $[x_0, x_1, ..., x_T]$, the prediction task requires us to progressively extend the initial condition $X(t_0) = x_0$ to $[X(t_0) = x_0, X(t_1) = x_1, ..., X(t_T) = x_T]$. This extension is necessary to align the expanded time-like dimension with the natural time sequence. However, all solutions of Neural ODEs are determined by the initial state $X(t_0)$. Once the network parameters θ are learned, there is no opportunity to optimize the dynamics over time.

Fortunately, following Neural ODEs, NCDE have emerged as another significant advancement in combining neural networks with dynamic systems. Let $0 < t_0 < T$ and $w, v \in \mathbb{N}$, consider $X : [t_0 : T] \to \mathbb{R}^w$ as a continuously mapped function of bounded variation. Let $f : \mathbb{R}^v \to \mathbb{R}^{v\times w}$ be Lipschitz continuous. If $y(t_0) = y_0$, and:

$$y(T) = y(t_0) + \int_{t_0}^{T} f(y(t))\mathrm{d}X(t), \quad \text{for } t \in (t_0, T] \quad (3)$$

The continuous path $y : [t_0 : T] \to \mathbb{R}^v$ describes the process of solving the controlled differential equation since it is controlled or driven by X. Here, $\mathrm{d}X(t)$ refers to the Riemann-Stieltjes integral. Given that $x(t_0) \in \mathbb{R}^x$ and $f(y(t)) \in \mathbb{R}^{v\times w}$, $f(y(t))\mathrm{d}X(t)$ represents the product of a matrix and a vector. It can be interpreted as a function from the space of paths to the space of paths. The input is the path X, and the output is the path y. By constructing an appropriate f, NCDE can be utilized to compute specific functions controlled by it.

Comparing the formulas of ODEs and NCDE, it becomes clear that the strength of NCDE in time series prediction lies in its dynamic system being driven by the dynamic process. This feature enables the system to adapt to new data, providing hidden states with continuous dependencies on the observed data.

C. Graph Attention Network

Graph Attention Networks dynamically learn the association weights between each node and its neighboring nodes, allowing for a more flexible capture of nonlinear relationships and dynamic changes in spatio-temporal sequences. At the same time, they pay more attention to neighboring nodes relevant to the current node, reducing issues related to interference from the overall structure present in traditional methods. Graph Attention Networks revolve around the attention mechanism. Let's assume we have a graph $G = (V, E)$, where V is the set of nodes, and E is the set of edges. Each node i has a feature representation $h_i^{(0)}$, which can initially be the input feature of the node. GAT introduces the attention mechanism, allowing each node to assign different attention weights to its neighboring nodes. Considering the attention weight between node i and node j, denoted as e_{ij}, the calculation is as follows:

$$e_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{a}^T[Wh_i \,\|\, Wh_j]\right)\right)}{\sum_{k\in N_i}\exp\left(\mathrm{LeakyReLU}\left(\vec{a}^T[Wh_i \,\|\, Wh_k]\right)\right)} \quad (4)$$

Where W is the weight matrix for linear transformation, $\vec{a}$ is the shared attention parameter vector, N_i is the set of neighboring nodes for node i, and $\|$ denotes vector concatenation. After obtaining the attention weights e_{ij}, the next step involves using these weights to perform a weighted average of the features of neighboring nodes to obtain the aggregated feature representation $h_i^{(l+1)}$:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N_i} e_{ij} W^{(l)} h_j^{(l)}\right) \quad (5)$$

Here, l represents the index of the layer, and σ is the non-linear activation function. By introducing the attention mechanism, GAT can dynamically learn the relationships between nodes, allowing each node to be assigned different weights during information aggregation. This capability enables GAT to better capture the complex dependencies between nodes when processing graph data.

IV. Model

A. Adjacency Matrix of Station

Constructing adjacency graphs is essential for accurately predicting passenger flow, capturing both the spatial relationships and mutual influences between stations. While researchers commonly measure straightforward connectivity using actual distances or subway network connections between stations, this method overlooks the more complex

passenger dynamics and functional similarities. Such elements are essential for understanding and predicting nuanced patterns of passenger flow.

To address this issue, this paper expands upon commonly used route connection graphs by integrating passenger flow similarity between stations to describe their spatial relationships more accurately. We assume a total of N stations for prediction. The graph structure is denoted as $G(V, E, W)$, where V represents the nodes corresponding to the stations, $|V| = N$. The edges of the graph, denoted by E, illustrate the associations between different stations, and W represents the weights of these adjacency edges. For stability during training, these weights are typically initialized to 0 or 1, indicating whether a relationship exists between the corresponding stations. The adjacency matrix used in training is represented as $A \in \mathbb{R}^{N\times N}$. Specifically, this study establishes two distinct graph structures: the route adjacency graph A_{Route} and the similarity graph A_{EMD} based on Earth Mover's Distance (EMD). These graphs are used for two independent graph learning modules.

The route adjacency graph A_{Route} is constructed from actual rail transit connections, providing a straightforward representation of low-order associations. If there is a direct connection between stations a and b in the real rail transit network, then $A_{a,b} = 1$; otherwise, $A_{a,b} = 0$. To prevent the adjacency matrix from neglecting self-connections, we set all diagonal elements of the adjacency matrix A to 1. This method can partially express the significant associations between stations.

However, in specific scenarios such as peak passenger flow periods, station connections should consider both route associations and passenger flow distribution similarity. For example, when two subway stations serve as commuting origins and destinations, their passenger flow patterns often correlate temporally. Establishing edges between these stations, even without direct routes, is crucial for effective graph learning models.

To address this, we construct a globally correlated adjacency graph using Earth Mover's Distance (EMD). By calculating EMD distances from historical passenger flow data, we determine node similarity and build an adjacency graph that better captures spatio-temporal evolution patterns. Specifically, for a pair of data sequences from two rail transit stations Seq_1 and Seq_2, the distance between these station distribution sequences is calculated using the following equation:

$$\mathrm{EMD}(X,Y) = \min_{\tau\in\Gamma(Seq_1,Seq_2)} \sum_{i,j} \tau_{i,j} \times M_{i,j} \tag{6}$$

Where M is the distance matrix constructed using Euclidean distance, where $M_{i,j}$ represents the distance between X_i and Y_j. Here, $\tau \in \Gamma(X,Y)$ denotes the transition strategy from X to Y, represented by a matrix with dimensions $L_1 \times L_2$. Specifically, $\tau_{i,j}$ signifies the transition cost from the i-th element of X to the j-th element of Y. The set Γ encompasses all feasible transition strategies. For every feasible transition strategy i, EMD assert:

$$\sum_j \tau_{i,j} = S_{n_i} \tag{7}$$

Equation (7) signifies that the total amount of the transformed distribution in the i-th row is equal to the i-th element in X. This condition ensures that the sum of elements in the i-th row of the transformation strategy matrix τ equals the i-th element in X. This requirement plays a vital role in minimizing transportation costs while preserving the total transition quantity in X. Importantly, the EMD metric method remains unaffected by the realistic distances between traffic detection nodes, enabling simultaneous consideration of the distribution of rail transit passenger flow data. In the computation process, we can employ efficient linear programming solvers, such as Gurobi and CPLEX [23]. These solvers render EMD well-suited for handling complex nonlinear distribution sequences like passenger flow.

In order to ensure the effectiveness of adjacency information in the matrix, connections with similarity below a threshold are set to zero. Taking the EMD method as an example, the process of constructing the adjacency matrix is illustrated in Fig. 2. According to the above approach, the globally correlated adjacency matrix we constructed is defined by the following equation:

$$A_{a,b} = \begin{cases} \exp\left(R_E^{a,b}\right), & \text{if } \exp\left(R_E^{a,b}\right) \geq \partial \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

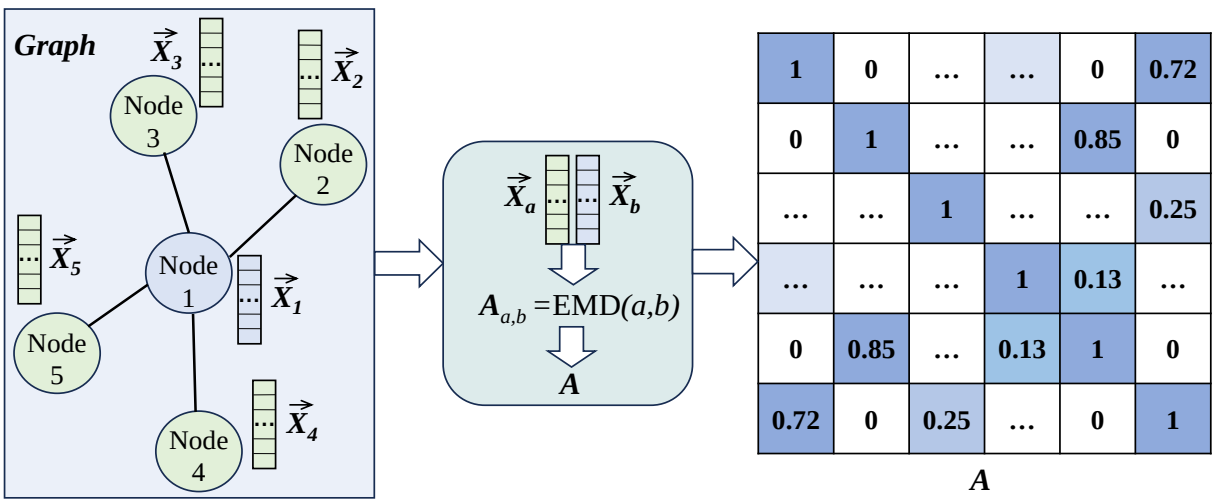


Fig. 2. Construction process of adjacency matrix

Where $R_E^{a,b}$ is the distance metric term, defined by the following equation:

$$R_E^{a,b} = -\frac{EMD_{a,b}^2}{\sigma^2} \tag{9}$$

The parameter σ in equation (9), along with the parameter ∂ defined in equation (8), functions as threshold parameters controlling the sparsity of the matrix. These parameters are pivotal in determining the strength and density of connections in the adjacency graph, allowing us to finely adjust the representation of correlation between traffic nodes.

Finally, symmetric normalization is applied to both adjacency matrices to ensure the stability and reliability of the training process, as outlined below:

$$A = D^{-1/2} A D^{-1/2} \tag{10}$$

Where D is the degree matrix, and its diagonal elements, represented by $D_{b,b}$ signify the outdegree of the nodes b .

B. Dual-Graph Attention Module

The fundamental concept of Graph Convolutional Networks (GCN) involves aggregating feature information from local neighboring nodes and updating the feature information of the current node. However, when predicting passenger flow in rail transit systems, where inter-station correlations are complex, pre-constructed adjacency graphs often fail to provide optimal feature representation for graph learning.

To address this limitation, this paper suggests assigning variable weights to nodes within the neighborhood, enhancing the capture of dynamic spatial dependencies. This study introduces two consecutive Graph Attention Networks tailored to the initially constructed adjacency matrices. These networks are designed to sequentially extract low-order and high-order spatial correlations between stations. Each layer of the graph attention network processes a graph input, starting with the graph structured for low-order relationships and then addressing high-order correlations. This approach ensures a thorough analysis of spatial dependencies, as illustrated in Fig. 3.

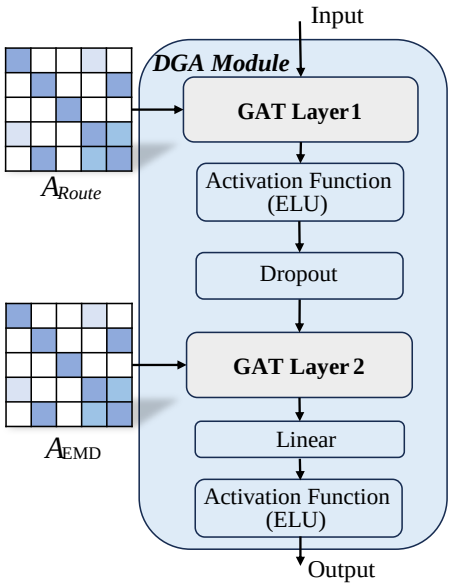


Fig. 3. The structure of Dual-Graph Attention

In the Graph Attention Layer, each station calculates attention coefficients for its neighboring nodes based on the weights between them, representing the focus of the current node on its neighbors. This functionality is implemented using a self-attention mechanism similar to a Transformer. The input to the Graph Attention Layer is the node features $X = [\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_N] \in \mathbb{R}^{N' \times F}$ and the adjacency matrices $A \in \mathbb{R}^{N' \times N'}$ for the two pre-constructed graphs, where N' is the number of stations and F is the number of features per node. The input features undergo a linear transformation to produce the feature set G for the GAT layer, where each element $\vec{g}_i$ represents the feature of the i-th node. This transformation is defined by the following equation:

$$\vec{g}_i = W \vec{X}_i \tag{11}$$

Here, $\vec{X}_i$ is the input feature of the node, W is the weight matrix.

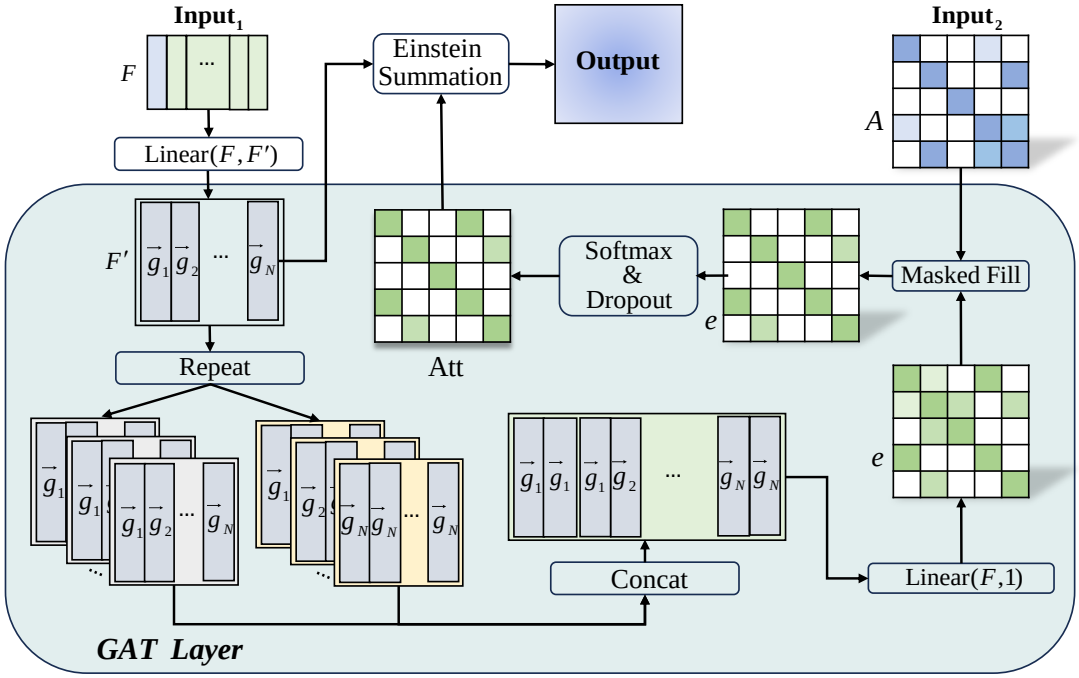


Fig. 4. Details of the Graph Attention Layer

To compute the source and target node features G_1 and G_2, the original features are replicated N times in two different dimensions, resulting in the following expressions:

$$G_1 = \left[\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_N, \ldots, \vec{g}_1, \vec{g}_2, \ldots, \vec{g}_N\right] \in \mathbb{R}^{N\times N\times F}$$
$$G_2 = \left[\vec{g}_1, \vec{g}_1, \ldots, \vec{g}_1, \ldots, \vec{g}_N, \vec{g}_N, \ldots, \vec{g}_N\right] \in \mathbb{R}^{N\times F\times N} \quad (12)$$

The source and target node features are paired and concatenated to form the following feature set:

$$G = \left[\vec{g}_1 \,\|\, \vec{g}_1, \vec{g}_1 \,\|\, \vec{g}_2, \ldots, \vec{g}_N \,\|\, \vec{g}_N\right] \in \mathbb{R}^{N\times N\times 2F} \quad (13)$$

Furthermore, attention weights between each pair of stations are computed through a linear mapping function $a: \mathbb{R}^{2F} \to \mathbb{R}^1$:

$$e_{i,j} = \mathrm{LeakyRelu}(a[g_i \,\|\, g_j]) \quad (14)$$

This process is illustrated in Fig. 4. Masked Fill denotes the element-wise masking operation. Specifically, for positions where the elements of the adjacency matrix A are equal to 0, the corresponding positions in the attention matrix e are set to negative infinity.

The Einstein Summation notation in Fig. 4. represents the aggregation operation in the graph. The features of the current station and its neighboring stations are jointly computed through the attention layer, resulting in adaptive weights. This ensures that the final node embedding prioritizes relevant neighboring nodes.

In predicting passenger flow in rail transit, the Dual-Graph Attention Module (DGA Module) effectively utilizes information from both the route adjacency graph and the global correlation graph.

The graph attention mechanism adaptively adjusts attention weights for different nodes. This means that the model can give more emphasis to geographical locations and global correlation information relevant to the prediction target. It adeptly handles the intricate structure of the traffic network, enhancing the model's robustness and suitability for the nonlinear characteristics of urban traffic systems.

C. Dual-Graph Attention Neural Controlled Differential Equation

The Dual-Graph Attention Network excels in mapping geographical and global connections but faces difficulties with time-series data. Its inability to effectively track the dynamic changes in passenger flow systems reduces its performance in scenarios involving nonlinear and fluctuating passenger volumes.

To overcome these challenges, we introduce Neural Controlled Differential Equations and seamlessly integrate them with the Dual-Graph Attention Network, creating a new model framework termed Dual-Graph Attention Neural Controlled Differential Equation (DGA-NCDE).

As outlined in Section III, NCDE offers a continuous modeling approach that addresses the limitations of RNNs' discrete structures in capturing complex dynamic changes. Adhering to the principles of NCDE, the dynamic change of the hidden state *h(t)* is described by a differential equation which is continuously controlled and adjusted by control signals, and this differential equation is parameterized by neural network. In this paper, the passenger flow sequence data is regarded as the control signal X_p changing with time, and its derivative describes the change dynamics of the time series data, and directly controls and drives the change of the hidden state in the NCDE, which is the core part of the differential equation solution. However, the passenger flow data are often discrete, and it is difficult to perform differential operations in the continuous domain of the series. To address this, we employ a natural cubic spline to create a continuous path, represented as X:

$$X = \mathrm{CubicSpline}(Xp) \quad (15)$$

This operation ensures that the model can obtain smooth and continuous control signals and derivatives when computing the differential equations. We define a set of discrete time point sequences $t=\{t_0, t_1, \ldots, T\}$ according to the time order of X, where T is the point in time at which we wish to solve for the last hidden state $h(T)$, t_0 is the initial time point and also the starting point of the evolution of the whole dynamic system. The value $X(t_0)$ of the continuous path X at t_0 generates the initial hidden state $h(t_0)$ through linear transformation:

$$h(t_0) = \mathrm{Linear}(X(t_0)) \quad (16)$$

In our model, DGA Module described in the previous section serves as the parameterized neural network g_θ: $\mathbb{R}^{W\times N'} \to \mathbb{R}^{W\times V\times N'}$ within the NCDE framework. It captures spatio-temporal dependencies and generates a vector field that represents the rate of change of the system state:

$$\mathrm{VectorField} = \mathrm{g}_\theta(h(t_i)) \quad (17)$$

Then, taking the derivative of the path X at the current time point t_i gives the path change rate $\mathrm{d}X(t)$ as follows:

$$\mathrm{d}X(t_i) = \mathrm{d}X / \mathrm{d}t\,|_{t=ti} \quad (18)$$

By combining the *VectorField* g and the path change rate dX(t), we get the change rate dh(t) of the hidden state h(t):

$$\mathrm{d}h(t_i) = \mathrm{VectorField}\cdot \mathrm{d}X(t_i) = \mathrm{g}_\theta(h(t_i))\cdot \mathrm{d}X(t_i) \quad (19)$$

Using a differential equation solver, we calculate the hidden state $h(t_i + \Delta t)$ after the initial time step by combining the current hidden state $h(t_i)$ and the change rate $\mathrm{d}h(t_i)$. The basic equation of its iterative calculation is as follows:

$$h(t_i + \Delta t) = h(t_i) + \int_{t_i}^{t_i+\Delta t} \mathrm{g}_\theta(h(t_i))\mathrm{d}X(t_i) \quad (20)$$

In this study, we describe the trajectory of the hidden state *h(t)* over the entire time domain as the hidden state path, or latent trajectory $h: [0, T] \to \mathbb{R}^{W\times N'}$. Based on equation (3), we define the DGA-NCDE model with the following equation:

$$h(T) = h(t_0) + \int_0^T g_\theta(h(t))\mathrm{d}X(t) \quad (21)$$

The hidden state *h(t)* represents the state of the dynamic system at time step *t*. Fig. 5 illustrates the DGA-NCDE framework, where boxes with solid lines indicate operational modules, and those with dashed lines depict feature representations. Fig. 5(A) provides detailed information about the neural network g_θ used within the NCDE. This includes a temporal feature extraction architecture set up prior to the DGA module. The architecture features two convolutional layers from a multilayer convolutional neural network, both with the same output dimensions. After processing through a non-linear activation function (Tanh), these branch features are fused through element-wise multiplication to form the input for the DGA module.

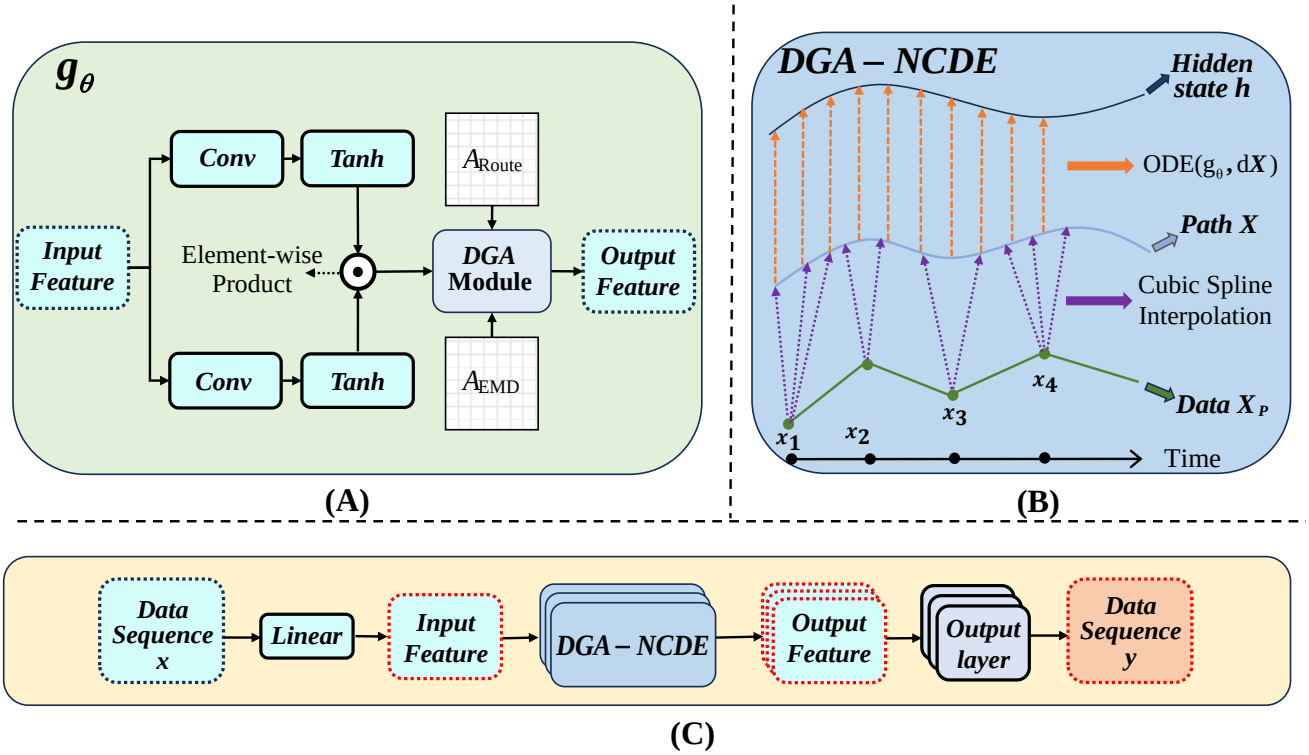


Fig. 5. Details of the DGA-NCDE framework

Fig. 5(B) demonstrates the NCDE's inference process. The cubic spline interpolation algorithm generates a set of cubic polynomials that define smooth, continuous intervals between each pair of data points. The dotted arrow represents the change of data flow in DGA-NCDE, and data X_p is the passenger flow sequence data of the station. For simplicity, the ODE solving process using g_θ and dX is defined as ODE(g_θ, dX), which is an iterative solving process. Methods like Euler and Runge-Kutta, among others, can be used to solve the hidden states at each time point. Fig. 5(C) presents a simplified schematic of the overall prediction framework, excluding basic operations such as regularization, concatenation, and transpose. The output layer in this diagram consists of linear layers.

To enhance the model's ability to extract deep features while remaining efficient, we developed two stacking patterns for the residual layers in the DGA-NCDE. The first pattern follows traditional residual layer stacking.

As shown in Fig. 6(A). If the length of the sequence to be predicted T' is greater than the threshold ω, the output features from the last residual layer are used as input for all subsequent output layers.This approach ensures that even with fewer layers, the model can capture long-term dependencies effectively. Conversely, for shorter sequences where $T' \le \omega$, we employ the residual recurrent layer stacking pattern depicted in Fig. 6(B). This configuration assigns each stacking block its own output layer, maximizing the utilization of features from each layer. In the experiments in this paper, the threshold ω is set to 4.

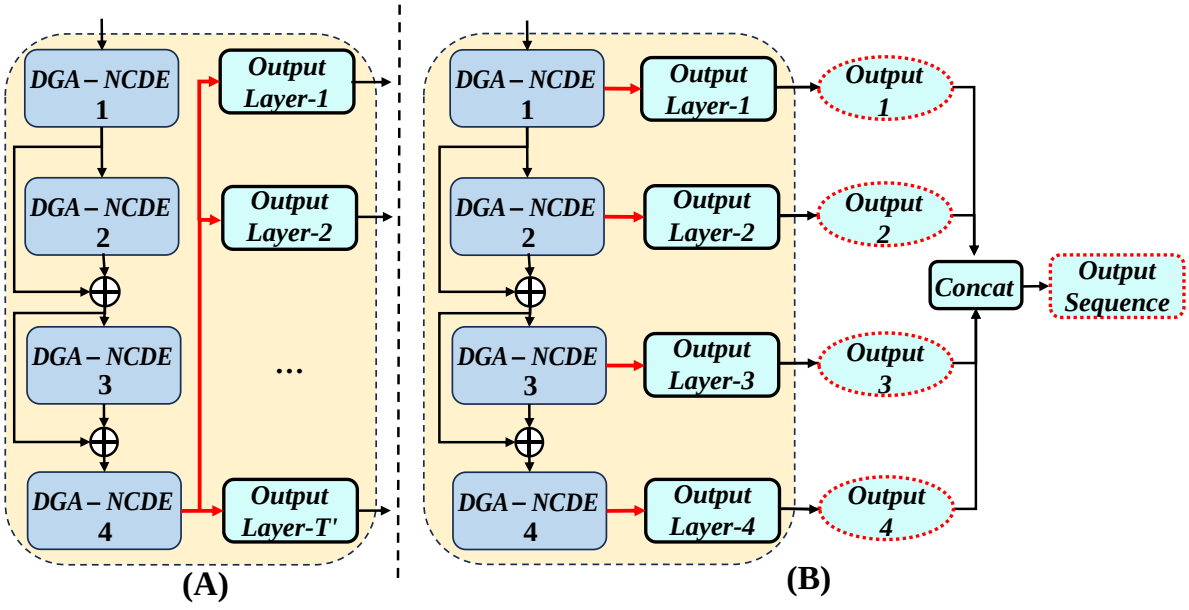


Fig. 6. Two residual cascade patterns of the output layer

D. Loss Function

To better balance sensitivity to outliers and robustness to noise, we choose to use the Huber Loss for our loss function, expressed as follows:

$$L_{\partial}\left(Y, f\left(X\right)\right)=\begin{cases}\frac{1}{2}\left(Y-f\left(X\right)\right)^{2} & \text{, if } \left|Y-f\left(X\right)\right|\le\partial\\ \partial\left|Y-f\left(X\right)\right|-\frac{1}{2}\partial^{2}\text{, if } \left|Y-f\left(X\right)\right|>\partial\end{cases} \tag{22}$$

Where Y represents the actual ground truth values, $f(X)$ denotes the model's predicted values, and ∂ is the hyperparameter of the Huber Loss, serving as the threshold for transitioning between square loss and absolute loss. When the absolute value of the prediction error is less than or equal to ∂, we apply square loss; when it exceeds ∂, the absolute loss function is employed.

V. Experiments

A. Datasets and Baselines

This paper rigorously evaluates the predictive capabilities of the DGA-NCDE model using two large-scale public datasets commonly used in the field: the Hangzhou Public Metro Smart Card dataset (HZMetro) and the Shanghai Metro Smart Card dataset (SHMetro). Detailed information about these datasets is provided in Table I:

TABLE I
Details of HZMetro and SHMetro

Datasets	HZMetro	SHMetro
Start date	Jan. 1st 2019	Jul. 1st 2016
End date	Jan. 25th 2019	Sept. 30th 2016
region	Hangzhou	Shanghai
Number of records	58.8 M	811.8 M
Average daily flow	2.35 M	8.8 M
Number of Stations	80	288

Notably, HZMetro originally included 81 stations, but one station lacked records, leaving 80 stations available for data prediction. To assess the performance of the proposed model, we conducted experimental trials comparing DGA-NCDE with the following baseline models:

1) Historical Average (HA) [24]: This method calculates passenger flow data for each station separately. It averages values for the same historical time period based on timestamps of historical time data to obtain predictions.
2) Long Short-Term Memory (LSTM) [25]: LSTM is An variant of RNN. It is a classic method for processing long sequences that possesses strong memory and long-term dependency modeling capabilities.
3) Gated Recurrent Unit (GRU) [26]: GRU is another excellent variant of RNN. It includes an update gate and a reset gate that work together to control the flow of information. GRU is often considered more trainable than LSTM and is frequently applied to sequence modeling tasks.
4) Deep Convolutional Recurrent Neural Network (DCRNN) [27]: This model applies dilated convolutions and gated recurrent units to extract spatial and temporal features. It is a classic method for spatio-temporal sequence prediction.
5) Graph WaveNet [28]: Graph WaveNet is inspired by WaveNet, which is a deep convolutional neural network used for speech synthesis. Graph WaveNet extends its ideas to the spatio-temporal graph domain and utilizes adaptive adjacency matrices for spatio-temporal feature extraction.
6) Spatio-Temporal Graph Convolutional Network (STGCN) [29]: STGCN introduces the concept of spatio-temporal graph convolution. This model effectively captures correlations between nodes in spatio-temporal data and dynamic changes in the temporal dimension, thereby enhancing its ability to model spatio-temporal information.
7) Spatio-Temporal Graph ODE Networks (STGODE) [30]: STGODE leverages graph neural networks and incorporates neural ordinary differential equations to model dynamic changes. It offers a continuous framework for implementing spatio-temporal graph residual networks.
8) Physical-Virtual Collaboration Graph Network (PVCGN) [21]: PVCGN constructs three distinct adjacency matrices. These matrices are designed to enhance the spatial extraction capabilities of graph convolution. It employs gated recurrent units to effectively capture the long-term spatio-temporal dependencies in rail transit passenger flow. This approach aims to improve the model's ability to discern complex patterns and relationships within the data.

B. Evaluation Metrics

During the experimental testing phase, our model is evaluated using three commonly used metrics for sequence prediction tasks: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics are defined by the following formulas:

Mean Absolute Error (MAE):

$$MAE=\frac{1}{n}\sum_{i=1}^{n}|Y_i-\hat{Y}_i| \tag{23}$$

Root Mean Square Error (RMSE):

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^2} \tag{24}$$

Mean Absolute Percentage Error (MAPE):

$$MAPE=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i-\hat{Y}_i}{Y_i}\right|\times 100\% \tag{25}$$

Where Y_i represents the actual values, $\hat{Y}_i$ represents the predicted values, and n represents the number of samples. The lower the values of the three evaluation metrics, the more accurate the model predictions. MAE is a standard measure of absolute error, while compared to MAE, RMSE emphasizes the overall trend of errors. On the other hand, MAPE is more reflective of error comparisons across datasets of different scales.

C. Experiment Settings

To ensure fairness in our experiments, we divided each of the two datasets into training, validation, and test sets in a ratio of 7:1:2. Each 15-minute interval serves as a timestep,

TABLE II
QUANTITATIVE COMPARISON OF DGA-NCDE AND 8 BASELINE MODELS ON THE HZMETRO DATASET

Time	Metric	HA	LSTM	GRU	DCRNN	Graph WaveNet	STGCN	STGODE	PVCGN	**DGA-NCDE**
15 min	MAE	36.37	26.73	25.69	23.76	24.07	23.85	23.75	23.28	**22.84**
	MAPE(%)	19.14	15.38	15.13	14.00	14.27	14.12	14.04	14.10	**13.90**
	RMSE	64.19	45.37	45.10	40.39	40.78	40.54	38.11	**38.66**	38.68
30 min	MAE	36.37	26.04	25.93	25.22	25.48	25.31	25.01	24.33	**23.25**
	MAPE(%)	19.31	15.61	15.35	14.99	15.23	14.93	14.47	14.21	**13.84**
	RMSE	64.10	45.37	45.26	42.57	42.80	42.50	41.21	39.94	**39.22**
45 min	MAE	36.23	25.30	26.36	26.97	27.15	27.06	26.34	24.62	**23.94**
	MAPE(%)	19.57	16.45	15.79	16.19	17.36	17.33	16.21	14.98	**14.13**
	RMSE	63.92	46.33	46.13	46.26	45.84	45.80	43.07	41.55	**41.28**
60 min	MAE	35.99	27.59	26.98	28.47	29.14	28.48	26.69	25.16	**24.71**
	MAPE(%)	20.01	18.26	17.20	18.16	19.37	19.30	17.42	15.97	**15.42**
	RMSE	63.72	47.81	47.69	49.35	49.89	48.77	44.82	42.46	**42.11**

and data from the previous hour are used to predict passenger flow for the next 15, 30, 45 minutes, and one hour. The final error metrics are calculated by averaging the errors in predicting both the incoming and outgoing passenger flows. All experiments were conducted on a Linux server powered by a 14 vCPU Intel® Xeon® Gold 6330 CPU at 2.00GHz and an NVIDIA RTX 3090 (24GB) GPU. We utilized the Runge-Kutta-4 method [31] as the ODE solver in the NCDE section. The feature embedding's hidden layer dimension is 64, and the stacking mode hyperparameter ω for the output layer is set to 4. For training, we employed the Adam optimizer coupled with a MultiStepLR strategy and set the initial learning rate to 0.012.

D. Overall Performance

We conducted experiments across four prediction time intervals to evaluate the performance of the DGA-NCDE model at various time granularities. The experimental results for HZMetro and SHMetro are presented in Tables II and III, respectively. These tables provide a comprehensive comparison of DGA-NCDE's performance across different prediction time horizons, with the best performances highlighted in bold and the second-best results underlined.

As evident in Tables II and III, the traditional time series forecasting algorithm HA exhibits relatively poor predictive performance across various time intervals. Its higher error values indicate limitations stemming from simple historical average calculations, rendering it ineffective in capturing the complexity and trend changes inherent in time series data. On the other hand, LSTM and GRU demonstrate superior predictive performance with lower errors. However, due to the inherent limitations of their iterative mechanisms, these models still face challenges in accurately predicting complex and time-varying data, particularly for long-term forecasts.

Models based on graph neural networks, such as DCRNN, Graph WaveNet, and STGCN, have demonstrated robust predictive performance across various time intervals. Leveraging the intricate combination of spatial and temporal features, these models outperform traditional methods. However, concerns arise about their stability over the long term. For example, Graph WaveNet exhibits relatively high error values when predicting passenger flow for 60 minutes, highlighting challenges in effectively capturing the complexity of time series data in certain cases within the

TABLE III
QUANTITATIVE COMPARISON OF DGA-NCDE AND 8 BASELINE MODELS ON THE SHMETRO DATASET

Time	Metric	HA	LSTM	GRU	DCRNN	Graph WaveNet	STGCN	STGODE	PVCGN	**DGA-NCDE**
15 min	MAE	48.26	26.68	25.91	24.04	24.91	24.37	24.01	23.51	**23.16**
	MAPE(%)	31.55	18.76	18.87	17.82	20.05	19.65	17.87	16.97	**16.90**
	RMSE	136.97	55.53	52.04	46.02	46.98	46.70	45.54	**44.38**	44.48
30 min	MAE	47.88	27.25	26.39	25.23	26.53	26.30	25.08	**24.36**	24.43
	MAPE(%)	31.49	19.04	19.20	18.35	20.38	19.78	18.70	18.23	**18.14**
	RMSE	136.81	57.37	54.02	49.90	51.64	50.08	48.24	**47.98**	48.04
45 min	MAE	47.26	28.08	27.17	26.76	28.78	27.43	26.04	25.42	**25.28**
	MAPE(%)	31.27	19.61	19.84	19.30	21.99	21.04	19.27	17.93	**17.73**
	RMSE	136.45	60.45	56.97	54.92	58.50	57.81	54.86	52.43	**51.26**
60 min	MAE	46.40	28.94	28.08	28.01	30.90	29.94	27.34	26.31	**25.95**
	MAPE(%)	30.80	20.59	21.03	20.44	24.36	22.25	20.21	18.52	**18.10**
	RMSE	135.72	63.41	59.91	58.83	65.08	60.37	56.30	54.93	**53.63**

spatio-temporal graph domain. STGCN may be influenced by the incompleteness of adjacency matrices when dealing with highly sparse graphs. While STGODE, with its continuous modeling of dynamic changes, excels in short-term predictions, its performance becomes average in long-sequence modeling scenarios.

The best-performing baseline model is PVCGN, which distinguishes itself by combining three different functional adjacency matrices to enhance the spatial extraction capabilities of graph convolution. However, in the context of large-scale complex networks, PVCGN may encounter challenges in computational efficiency, particularly during training and inference.

Leveraging the exceptional long-term spatial-temporal correlation extraction capabilities of the NCDE and graph attention modules within DGA-NCDE, although DGA-NCDE and PVCGN, the latter applying three graph structures, demonstrate similar performance in short-term predictions, the more lightweight DGA-NCDE exhibits lower errors in long-term predictions. In the experiments conducted on two public datasets, DGA-NCDE exhibits notable improvements compared to the optimal baseline PVCGN. Specifically, the Mean Absolute Error (MAE) of DGA-NCDE is reduced by 1.79% and 1.37%, while the Mean Absolute Percentage Error (MAPE) is reduced by 3.44% and 2.27%, respectively.

This underscores its inherent advantage in capturing long-term scale spatial-temporal features, thereby validating the exceptional performance of this method in subway passenger flow prediction tasks.

E. Efficiency Comparison of the Models

In this section, we will compare the model parameters, training time, and inference time of our model with other existing models using the HZMetro dataset. A detailed comparison of the parameter quantities and time consumption of different models is shown in Table IV. The training time represents the time taken per epoch with a batch size of 32, while the inference time refers to the duration for the model to infer the future one-hour passenger flow for all stations across the city.

TABLE IV
COMPARISON OF MODEL PARAMETERS AND TIME CONSUMPTION

Model	# Parameters	Trainging Time(s/epoch)	Inference Time (s)
LSTM	763168	3.96	0.00048
DCRNN	298242	23.69	0.01184
PVCGN	37598782	29.45	0.04591
STGODE	712200	19.57	0.01202
DGA-NCDE	422072	14.34	0.00586

As shown in Table IV, the DGA-NCDE model reduces the number of parameters by 98.88% and 40.73% in comparison to the PVCGN and STGODE models, respectively. Additionally, the DGA-NCDE model also demonstrates efficiency in time consumption, with a training time reduction of approximately 51.31% and 26.72%, and an inference time reduction of about 87.23% and 50.41% when compared to the PVCGN and STGODE models, respectively.

Moreover, Table IV reveals a discernible relationship between the model's parameter quantity and time consumption. However, this relationship is not strictly linear. Different model structures, algorithms, and optimization strategies can result in varying computational costs for the same number of parameters during training. For instance, the relatively simple structure of LSTM leads to fewer parameters, translating into significantly lower training times compared to other models that consider spatial dependencies. However, in practical scenarios, LSTM's prediction accuracy tends to be lower than that of more sophisticated models. DCRNN, despite having fewer parameters, exhibits a longer training time due to its intensive graph operations and direct application of recursive iterative calculations. This results in a deeper computational graph during training, necessitating higher computational resources. PVCGN, characterized by a relatively short training time, boasts a large number of parameters. While this empowers the model with robust fitting capabilities, it demands substantial computational resources and memory to handle the extensive parameter set.

In contrast, the proposed model DGA-NCDE, even with the utilization of two adjacency graphs, maintains a lower parameter count. This can be attributed, in part, to the integration of the graph attention network into NCDE, which contributed to the stabilization of gradient flow during training. Additionally, the parallel structure of the temporal feature extraction network and the output layer enhances the model's data processing speed.

F. Ablation experiment

To assess the effectiveness of the proposed method and its constituent modules, a series of ablation experiments were conducted on the HZMetro dataset. Four variant models for DGA-NCDE were specifically designed:

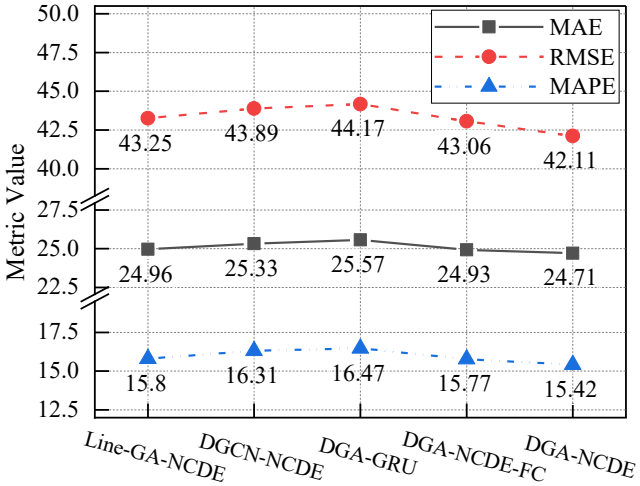


Fig. 7. Ablation experiments of DGA-NCDE

1) Line-GA-NCDE: In this variant, the adjacency matrix constructed by Earth Mover's Distance (EMD) distance is replaced with a subway line adjacency matrix.
2) DGCN-NCDE: This variant substitutes the graph attention network with a standard graph convolution, incorporating a convolutional layer depth of 3 layers.
3) DGA-GRU: Integrating the dual graph attention network into the Gated Recurrent Unit (GRU) architecture, this variant aimed to explore the effectiveness of dynamic spatio-temporal modeling in Neural Controlled Differential Equation.

4) DGA-NCDE-FC: This variant replaces the final output module with a standard fully connected layer, allowing for an evaluation of the effectiveness of the residual cyclic output layer in DGA-NCDE.

Fig. 7 illustrates the results of the ablation experiments, showcasing the performance of various variant models in comparison to the original DGA-NCDE. Notably, among all the variant models, DGA-GRU and DGCN-NCDE, considered the two most impactful variants, exhibit the largest performance degradation in model predictions. The diminished predictive performance of DGA-GRU underscores the challenges associated with adopting non-dynamic spatio-temporal relationship modeling methods in passenger flow prediction. On the other hand, the DGCN-NCDE variant shows an increase in Mean Absolute Error (MAE) from 24.71 to 25.33. This further emphasizes that the combination of graph attention networks and NCDE dynamic modeling proves more adept at leveraging graph structural information compared to graph convolutional networks. This adaptability enables the model to flexibly adjust to the spatio-temporal dependency relationships between subway stations.

While the prediction performance of the other two variants has relatively little impact, there remains a noticeable gap when compared to the original DGA-NCDE. Collectively, these experimental results robustly affirm the effectiveness of the module methods integrated into DGA-NCDE.

G. Case Study

The DGA-NCDE model presents a novel approach by integrating geographical connections and similarities in passenger flow patterns, thereby capturing a comprehensive context for predictive capabilities. As a result, this model excels in accounting for both peak flows in commercial areas and regular flows in residential areas.

To demonstrate the accuracy and applicability of the DGA-NCDE model in various railway station settings, we conducted real-world experiments on passenger inflow and outflow over a single day. Specifically, we focused on Stations #5, #27, and #75, chosen as representative urban rail transit hubs.

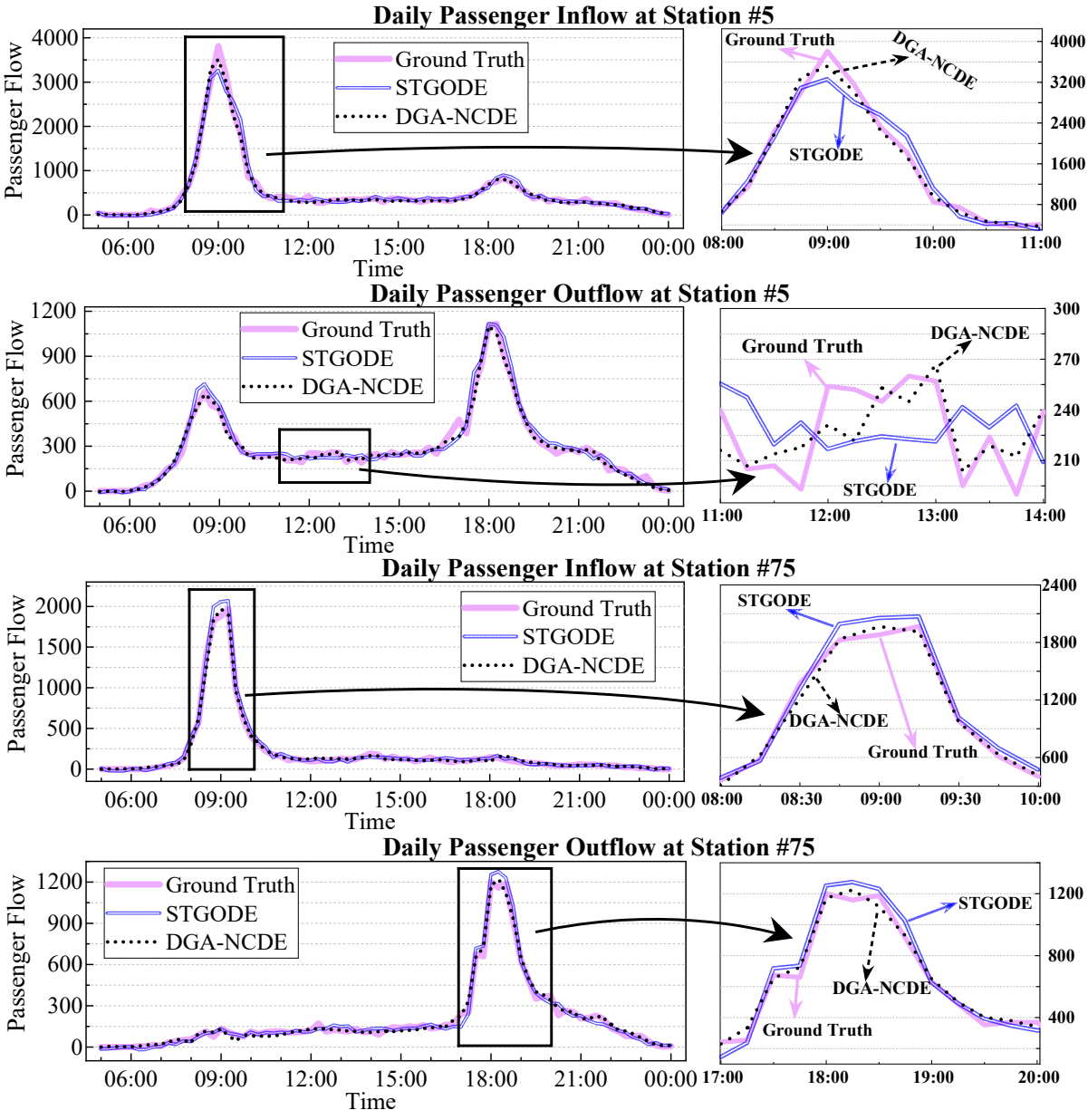


Fig. 8. Comparison of passenger flow prediction performance at high daily traffic stations #5 and #75

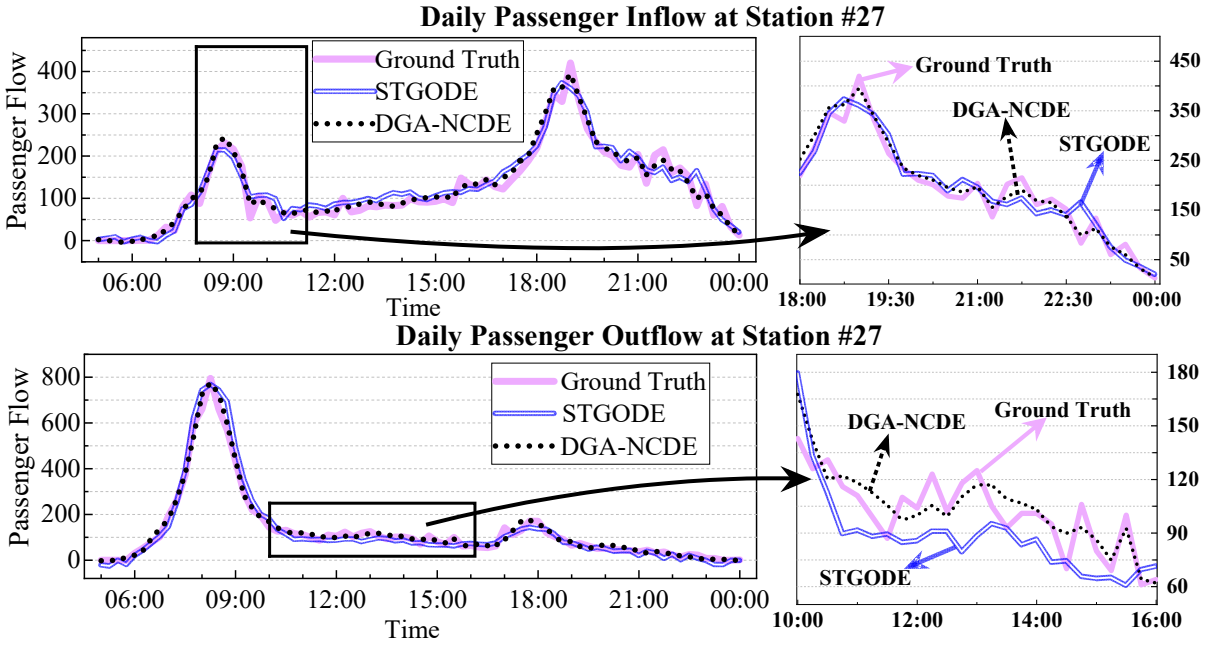


Fig. 9. Comparison of passenger flow prediction performance at low daily traffic station #27

As comparing the empirical data collected from these stations against the predictive outcomes generated by the STGODE model, which employs Neural ODEs, we sought to assess the efficacy of the DGA-NCDE model. The selected stations exhibit typical urban rail transit characteristics: Stations #5 and #75 show high-traffic, single-peak patterns during rush hours, while Station #27, although lower in average daily traffic, displays mild peaks twice a day, indicating a multi-fluctuating traffic pattern.

In Fig. 8, noteworthy surges in traffic volumes are discernible during the morning and evening rush hours at Stations #5 and #75. The DGA-NCDE model not only captures these peak values accurately but also precisely models the rates of traffic increase and decrease, displaying superior performance to that of the STGODE model. The latter exhibits marginal delays in peak prediction and trend accommodation, attributable to its reliance on inputting the initial state of the entire historical sequence into a fixed dynamic system for inferring future states, which potentially constrains its capacity to predict outcomes in intricate sequences. In contrast, the core of the NCDE technology lies in its control function, which enables a continuous influence of historical information on the evolutionary trajectory of the system. This flexibility empowers the model to dynamically adjust its impact in response to actual changes in the input data rather than evolving along a predetermined path. Such an approach is particularly crucial during periods of rapid traffic changes and continually enhances the model's efficiency and accuracy in using historical data.

Fig. 9 further affirms the prowess of the DGA-NCDE model in handling complex temporal patterns through its dual-peak traffic predictions at Station #27. The model clearly differentiates and accurately forecasts the traffic changes between the two peaks, proving its adaptability and stability in variable environments.

VI. Conclusion

The accurate prediction of urban rail transit behaviors is a fundamental aspect of effective traffic engineering in the context of smart cities. The efficacy of such predictions holds profound implications for the operational efficiency and overall functionality of urban transportation systems. However, traditional prediction models have often been hampered by their reliance on static modeling approaches, particularly in the utilization of graph neural networks for capturing spatio-temporal correlations within urban rail transit networks. This static approach has shown limitations in fully encapsulating the inherent dynamics of transportation systems and the intricate interdependencies within passenger flow patterns, thereby constraining the capacity of deep learning models to develop a holistic comprehension of urban rail transit systems.

Therefore, this study proposes a pioneering predictive model termed Dual Graph Attention-NCDE (DGA-NCDE). The model takes full advantage of two adjacency graphs constructed based on geographic line associations and Earth Mover's Distance (EMD) associations to extract global spatio-temporal correlations among subway stations. Subsequently, these two adjacency graphs are input into two consecutive graph attention layers to sequentially extract low- to high-order spatio-temporal correlation features. Finally, the constructed modules are integrated into the continuous modeling of NCDE, and prediction values of passenger flow are obtained through a carefully designed residual cyclic output layer.

This research expands the solution for rail transit passenger flow prediction to include dynamic modeling with NCDE. The combination of NCDE and graph attention neural networks builds an efficient and lightweight model based on spatio-temporal correlations in subway passenger flow prediction. By harnessing the correlation learning ability of graph attention and the powerful continuous dynamic modeling capability of NCDE, a new feasible solution for spatio-temporal prediction of rail transit passenger flow is provided. Extensive experimental analyses demonstrate that the proposed DGA-NCDE model outperforms the best baseline models on the two datasets. Notably, the MAPE in long-term prediction tasks using the

HZMetro dataset is reduced by 3.44% compared to existing models. Simultaneously, the proposed model achieves significant enhancements in terms of parameter efficiency and inference speed. Subsequent supplementary experiments further explain the roles of each module and the effectiveness and efficiency of the methods.

In summary, this research endeavors to offer a novel perspective on predicting passenger flow trends, with the outcomes of our experiments showcasing a marked enhancement in efficacy and accuracy wrought by the adoption of dynamic modeling techniques. The utilization of spatio-temporal features in our approach enables the construction of streamlined prediction models with reduced parameter complexity. Notwithstanding these advancements, there remains ample scope for refining the prediction of urban rail transit passenger flows. This may include a comprehensive exploration of dynamic modeling methodologies, as well as innovative applications and extensions of graph learning techniques to extract more versatile spatio-temporal information for intricate predictions. Future endeavors will concentrate on delving deeper into the optimization of passenger flow forecasting, underscoring our commitment to furnishing cutting-edge solutions and inventive strategies for the evolution of sophisticated smart city ecosystems.

REFERENCES

[1] J. Wang, R. Wang, and X. Zeng, "Short-term passenger flow forecasting using CEEMDAN meshed CNN-LSTM-attention model under wireless sensor network," *IET Communications*, vol.16, no.10, pp. 1253-1263, 2022.

[2] Q. Wang, X. Chen, C. Zhu, K. Zhang, R. He, and J. Fang, "Short-term Traffic Flow Prediction Based on Spatiotemporal and Periodic Feature Fusion," *Engineering Letters*, vol.32, no.1, pp. 43-58, 2023.

[3] C. Liu, H. Dai, S. Wang, and J. Chen, "Remote Sensing Image Scene Classification Based on Multidimensional Attention and Feature Enhancement," *IAENG International Journal of Computer Science*, vol.50, no.4, pp. 1337-1346, 2023.

[4] X. P. Chen, and Y. Xu, "A Multi-Dimensional Attention Feature Fusion Method for Pedestrian Re-identification," *Engineering Letters*, vol.31, no.4, pp. 1365-1373, 2023.

[5] P. Du, J. Wu, C. Ma, H. Hu, Y. Chen, and J. Li, "A Graph Contrastive Learning with Feature Perturbation for Recommender Systems," *IAENG International Journal of Computer Science*, vol.50, no.4, pp. 1368-1376, 2023.

[6] A. Nejadettehad, H. Mahini, and B. Bahrak, "Short-term Demand Forecasting for Online Car-hailing Services Using Recurrent Neural Networks," *Applied Artificial Intelligence*, vol.34, no.9, pp. 674-689, 2020.

[7] J. Chen, H. Dai, S. Wang, and C. Liu, "Improving Accuracy and Efficiency in Time Series Forecasting with an Optimized Transformer Model," *Engineering Letters*, vol.32, no.1, pp. 1-11, 2023.

[8] P. Kidger, J. Morrill, J. Foster, and T. Lyons, "Neural controlled differential equations for irregular time series," *34th Conference on Neural Information Processing Systems*, pp. 6696-6707, 2020

[9] P. Velikovi, A. Casanova, P. Lio, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," *6th International Conference on Learning Representations (ICLR)2018*, pp. 1-12, 2018

[10] Y. Li, H. Han, X. Liu, and C. Li, "Passenger flow forecast of Sanya airport based on ARIMA Model," *4th International Conference of Pioneer Computer Scientists, Engineers and Educators, ICPCSEE 2018, September 21, 2018 - September 23, 2018*, pp. 442-454, 2018

[11] C. Ding, J. Duan, Y. Zhang, X. Wu, and G. Yu, "Using an ARIMA-GARCH Modeling Approach to Improve Subway Short-Term Ridership Forecasting Accounting for Dynamic Volatility," *IEEE Transactions on Intelligent Transportation Systems*, vol.19, no.4, pp. 1054-1064, 2018.

[12] L. Mou, P. Zhao, H. Xie, and Y. Chen, "T-LSTM: A Long Short-Term Memory Neural Network Enhanced by Temporal Information for Traffic Flow Prediction," *IEEE Access*, vol.7, pp. 98053-98060, 2019.

[13] J. Guo, Z. Xie, Y. Qin, L. Jia, and Y. Wang, "Short-Term Abnormal Passenger Flow Prediction Based on the Fusion of SVR and LSTM," *IEEE Access*, vol.7, pp. 42946-42955, 2019.

[14] W. Zhao, Y. Gao, T. Ji, X. Wan, F. Ye, and G. Bai, "Deep Temporal Convolutional Networks for Short-Term Traffic Flow Forecasting," *IEEE Access*, vol.7, pp. 114496-114507, 2019.

[15] S. Sha, J. Li, K. Zhang, Z. Yang, Z. Wei, X. Li, et al., "RNN-Based Subway Passenger Flow Rolling Prediction," *IEEE Access*, vol.8, pp. 15232-15240, 2020.

[16] V. N. Katambire, R. Musabe, A. Uwitonze, and D. Mukanyiligira, "Forecasting the Traffic Flow by Using ARIMA and LSTM Models: Case of Muhima Junction," vol.5, no.4, pp. 616-628, 2023.

[17] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, "Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.11, pp. 7004-7014, 2021.

[18] H. Peng, H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, et al., "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Information Sciences*, vol.521, pp. 277-290, 2020.

[19] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, and B. Yin, "Metro Passenger Flow Prediction via Dynamic Hypergraph Convolution Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.12, pp. 7891-7903, 2021.

[20] X. Yang, Q. Xue, M. Ding, J. Wu, and Z. Gao, "Short-term prediction of passenger volume for urban rail systems: A deep learning approach based on smart-card data," *International Journal of Production Economics*, vol.231, pp. 107920, 2021.

[21] L. Liu, J. Chen, H. Wu, J. Zhen, G. Li, and L. Lin, "Physical-Virtual Collaboration Modeling for Intra- and Inter-Station Metro Ridership Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.4, pp. 3377-3391, 2022.

[22] S. Wang, H. Dai, L. Bai, C. Liu, and J. Chen, "Temporal Branching-Graph Neural ODE without Prior Structure for Traffic Flow Forecasting," *Engineering Letters*, vol.31, no.4, pp. 1534-1545, 2023.

[23] T. A. Ademoye, A. Davari, C. C. Castello, S. Fan, and J. Fan, "Path planning via CPLEX optimization," *40th Southeastern Symposium on System Theory*, pp. 92-96, 2008

[24] G. Chang, Y. Zhang, D. Yao, and Y. Yue, "A summary of short-Term traffic Flow forecasting methods," *11th International Conference of Chinese Transportation Professionals: Towards Sustainable Transportation Systems*, pp. 1696-1707, 2011

[25] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *29th Annual Conference on Neural Information Processing Systems*, pp. 802-810, 2015

[26] K. Cho, B. Van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103-111, 2014

[27] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *6th International Conference on Learning Representations, (ICLR) 2018*, pp. 361-376, 2018

[28] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for Deep Spatial-Temporal Graph Modeling," *The 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1907-1913, 2019

[29] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pp. 3634-3640, 2018

[30] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting," *27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2021*, pp. 364-373, 2021

[31] W. W. Hager, "Runge-Kutta methods in optimal control and the transformed adjoint system," *Numerische Mathematik*, vol.87, no.2, pp. 247-282, 2000.