Research Article

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Transportation Research Record
1–21

DOI: 10.1177/03611981241248160
journals.sagepub.com/home/trr
Sage

Short-Term Origin-Destination Demand Prediction Based on Spatiotemporal Encoder-Decoder Network with a Residual Feature Extractor

Xiaohui Zhong[1], Jinlei Zhang[1], Qiang Hua[2], Lixing Yang[1], and Ziyou Gao[1]

Abstract

Online ride-hailing services play a crucial role in daily transportation, However, challenges persist in certain regions with limited access, and drivers encounter difficulties in receiving orders. Accurate prediction of short-term origin-destination (OD) demand is crucial for addressing these issues. This study leverages recent advancements in artificial intelligence and big data to introduce a spatiotemporal encoder-decoder network with a residual feature extractor (RF-STED) for short-term OD demand prediction in online ride-hailing services. The RF-STED model, built on deep learning models such as graph convolutional networks and convolutional long short-term memory (Conv-LSTM), includes spatiotemporal networks, encoding layers, and a residual feature extractor. The spatiotemporal network has two branches: branch one processes multi-pattern OD data using a multi-pattern temporal feature extraction module, utilizing a multi-channel Conv-LSTM to capture temporal correlations. Branch two utilizes a multi-spatial feature extraction module to convert OD pair associations into a spatial topology, extracting multi-spatial correlations. The encoding layer captures spatiotemporal dependencies, while the residual feature extractor decodes compressed vectors back into an OD graph for forecasting future demand. Experiments with a Manhattan taxi dataset in the U.S. show the RF-STED model outperforms 10 baseline models and four ablation models. The results emphasize the model's strength and robustness in short-term OD flow prediction.

Keywords

online ride-hailing service, short-term OD demand prediction, deep learning, spatiotemporal feature

The expansion of the internet in 2010 led to the emergence of various transportation network companies such as Uber, Lyft, and Didi, introducing services such as as bike-sharing and online ride-hailing. Over time, online ride-hailing and bike-sharing services have developed and matured. According to data, in 2020, more than 24 countries and over 700 cities worldwide had embraced online ride-hailing and similar services (*1*). In China, Didi Travel took the lead in online ride-hailing services, leading to a rapid proliferation of other online ride-hailing apps. Online ride-hailing services have become prominent because of their 24/7 availability, speed, and transportation efficiency. They have become a crucial mode of transportation in the daily lives of people, especially in large and medium-sized cities, where they are considered essential modes of transportation.

However, for online ride-hailing services, challenges persist in certain areas where issues such as unavailable rides or excessive demand may result in lower passenger satisfaction levels. In the context of online ride-hailing, critical factors include the dispatch speed of the platform

[1]School of Systems Science, Beijing Jiaotong University, Beijing, China
[2]Hebei Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding, PR China

Corresponding Author:
Jinlei Zhang, zhangjinlei@bjtu.edu.cn

and the responsiveness of drivers to incoming ride requests. Consequently, addressing the temporal and spatial imbalances in supply and demand within the operations of online ride-hailing services becomes a crucial and intricate challenge (*2*). Against this backdrop, this study aims to uncover patterns and trends in online ride-hailing demand across various regions and time frames. The objective is to enhance the overall passenger experience in online ride-hailing services by establishing a scientifically sound ride-hailing dispatch system. Additionally, accurate origin-destination (OD) demand information enables efficient dynamic pricing schemes to achieve higher platform profit (*3*).

The ride-hailing demand prediction problem includes region-level prediction and OD-level prediction (*4*). At the region-level prediction, the objective is to predict aggregate trip generations and attractions for each region. Various prediction methods have been proposed in the literature, including traditional time series forecasting and machine learning methods (*5–9*). Most of these studies focus on predicting traffic generation and attraction for each region. They typically partition the study area into some regular regions, and then pairwise relationships between regions are mathematically modeled through Euclidean structures. Nevertheless, ride-hailing data involves non-Euclidean correlations, and modeling such correlations is crucial for accurate predictions (*10*). Grid-based segmentation methods cannot capture these non-Euclidean relationships. As a result, some studies have begun to capture non-Euclidean pairwise relationships between different regions. The concept of OD-level prediction has been proposed.

OD-level prediction is more challenging than region-level prediction. At the OD prediction level, OD pairs are often treated as nodes in a graph structure. The static relationships between nodes (such as spatial distance, traffic volume similarity, and functional similarity) are used as edges. Mining the spatiotemporal dependencies between OD pairs becomes a key challenge. In recent studies, researchers have initially considered time and space separately. They individually explore the temporal and spatial correlations of OD demand. Using encoding principles, they encode temporal and spatial correlations into a dense vector. With a specific decoder, it can mine spatiotemporal correlations and ultimately make OD predictions (*1, 11, 12*).

However, the encoder-decoder structure of existing studies is relatively simple in the design of the decoding module, which imposes limitations on exploring spatiotemporal correlations. Building on previous research, this paper introduces a novel deep-learning framework called the spatiotemporal encoder-decoder network with a residual feature extractor (RF-STED). It is based on a spatiotemporal encoding-decoding architecture. In the encoding module, we adopted a common approach, similar to most literature, by utilizing graph convolutional networks (GCN) to extract spatial dependencies of OD demand and long short-term memory (LSTM) to capture temporal dependencies of OD demand. What distinguishes our model from conventional approaches is the parallel connection of multiple GCNs to extract diverse spatial dependencies. Simultaneously, our model utilizes convolutional LSTM (Conv-LSTM) network to maintain the intrinsic format of OD data, and multiple Conv-LSTM (MConv-LSTM) networks are parallelly connected to capture diverse temporal dependencies under different time patterns. This approach enhances the flexibility and applicability of our model. In the decoding module, we adopted a more complex design than existing literature. The carefully designed decoder facilitates a better extraction of spatiotemporal correlations in OD demand. The contributions of this paper are outlined as follows.

Firstly, this paper incorporates the trend relationship graph among OD pairs to depict spatial relationships between them. Building on this, a classification study of OD pairs is conducted using K-means clustering. The results demonstrate that distinct spatial relationships indeed exist among different OD pairs, and the inclusion of this spatial relationship within the deep learning framework leads to enhanced predictive accuracy of the model.

Secondly, this paper proposes a new feature extraction module—residual feature extractor—which uses convolutional check vectors of different sizes for feature extraction. The results show that the residual feature extractor performs well in the feature extraction of encoding vectors.

Thirdly, a novel deep-learning model is proposed in this paper. The model is capable of uncovering various spatial correlations between OD pairs and multiple temporal patterns inherent in OD pairs. It also features a well-designed encoding and decoding scheme to synergistically capture both temporal and spatial correlations. The model exhibits strong advantages.

Lastly, leveraging an online ride-hailing dataset from the Manhattan area in the U.S., benchmark model experiments and ablation studies are conducted. The outcomes reveal that the proposed model outperforms other approaches, demonstrating superior predictive performance and applicability.

Literature Review

OD Demand Predictions

The issue of predicting OD demand has been a focal point of research among scholars both domestically and internationally, yielding significant breakthroughs. The

evolution of OD demand prediction has undergone three distinct phases, with each stage marked by distinctive contributions and advancements. These stages constitute the trajectory through which OD flow prediction methodologies have evolved, reflecting the dynamic progress and innovative endeavors within this field of study. As researchers continue to delve into the intricacies of OD flow prediction, a comprehensive understanding of its development will offer valuable insights into enhancing the efficiency and effectiveness of ride-hailing services, ultimately improving the quality of urban mobility for passengers (*13*).

In the first stage, the majority of OD studies were primarily based on long-term OD prediction. One of the most classic models is the four-stage model proposed in 1962 by the Chicago Area Transportation Study in the U.S. (*14*). It includes traffic generation, traffic distribution, modal split, and traffic assignment. Here, traffic generation represents the amount of traffic a region will generate within a certain period; traffic distribution represents the traffic volume from one region to another within a specific time; modal split indicates the travel volume of different modes of transportation from one region to another; and traffic assignment involves allocating the specific traffic volume from one region to another into the road network. Long-term OD prediction is mainly used for urban planning and development (*15*).

In the second stage, because of the requirements for real-time and accurate short-term OD flow prediction, researchers began to use traditional mathematical models for short-term OD flow prediction (*16*). Initially, to capture the temporal dependencies of individual OD pairs at different time intervals, some time series methods were developed to predict traffic demand in specific regions, including autoregressive integrated moving average (ARIMA) and Markov models (*5*, *6*). Yao et al. proposed a nonlinear programming model based on the least squares method to predict dynamic OD matrices using historical automatic fare collection data (*17*). This model integrates more traffic features. Yao et al. utilized a state-space approach to construct a short-term passenger flow OD estimation model applicable to urban rail transit networks (*18*). Traditional models performed unsatisfactorily with significant errors. Modeling complexity and feasibility were low in short-term OD flow prediction.

In the third stage, the rise of machine learning and deep learning in the field has brought vitality to this research area (*19*, *20*). In general, the establishment of machine learning and deep learning models is mainly considered from two aspects: firstly, the temporal correlation between OD flows and, secondly, the spatial correlation between OD flows. With regard to modeling temporal features, in 2015, the deep learning model LSTM was first applied in the field of traffic prediction (*21*). Researchers found that deep learning models perform well in exploring the temporal correlation of OD demand. Jiang et al. utilized the LSTM model for OD flow prediction, training a separate LSTM model for each region and combining multiple LSTM models to predict OD flows for all regions (*22*). Subsequently, researchers began exploring more complex time series prediction deep learning models. The temporal convolutional network has been proven to be more stable during training and has the ability to parallelize effectively with appropriate adjustments (*23*, *24*). These advantages are particularly beneficial when dealing with large-scale demand prediction problems.

In the context of spatial feature modeling, the primary consideration is the partitioning of the study area. Initially, CNN was the first network used to extract spatial dependencies (*25*). Liu et al. applied the convolutional neural network (CNN) to predict road traffic OD flows (*26*). They divided the entire city into rectangular regions, using a grid-based approach where the spatial structure is organized in matrix form. In this case, pairwise relationships between regions were mathematically modeled through Euclidean structures, which is suitable for traditional CNN operations. However, ride-hailing data involves non-Euclidean correlations, and modeling such correlations is crucial for accurate predictions (*9*). Grid-based segmentation methods cannot capture these non-Euclidean relationships. The emergence of GCN provides a solution to this problem, allowing researchers to customize graph structures to effectively capture spatial information from non-Euclidean structured data. In the field of transportation, Chai et al. proposed a multi-graph CNN model to predict station-level bicycle traffic, customizing three graph structures: distance graph, interaction graph, and association graph, effectively capturing various spatial information (*27*). Sun et al. introduced a multi-view GCN to predict the inflow and outflow of irregular areas (*28*). They proposed a new spatiotemporal dependency generative adversarial network, utilizing multi-GCN to explore and model heterogeneous correlations between sensors from multiple perspectives, facilitating the effective extraction of deep non-Euclidean spatial features.

In recent years, with the rapid development of deep learning, researchers are no longer confined to modeling from only the temporal or spatial dimension but are integrating both aspects to explore the spatiotemporal features among OD demand. Initially, Shi et al. redefined the hidden layer of LSTM, incorporating convolutional operations and proposing the conv-LSTM module to simultaneously learn spatiotemporal dependencies by integrating CNN with LSTM (*29*). Jiang et al. combined LSTM and CNN to predict the OD flows of bike-sharing

systems (*30*). Their model simultaneously considers the temporal and spatial correlations of OD demand, leading to further improvement in predictive accuracy. Yu et al. developed the spatiotemporal GCN to effectively capture spatiotemporal information from non-Euclidean structured data (*31*, *32*). Building on Yu et al., Geng et al. introduced the spatiotemporal multi-GCN (MGCN) model for predicting region-based ride-hailing demands (*10*, *31*). The model initially represents non-Euclidean relationships between regions by designing various adjacency matrices. Subsequently, it captures temporal dependencies using a recursive neural network. In subsequent work, researchers further introduced lower-level grouped GCN and upper-level multilinear relation GCN to learn more generalized features.

Encoder-Decoder Architecture

In recent years, with the rapid advancement of deep learning and hardware upgrades, researchers have been able to train more complex models. However, models considering time and space separately merely explore temporal and spatial dependencies. To delve deeper into the spatiotemporal correlations among OD demand, an OD prediction model based on the encoder-decoder paradigm is proposed. Goyal et al. developed an encoder-decoder architecture based on LSTM and CNN (*11*). Subsequently, Ke et al. proposed a spatiotemporal encoder-decoder residual multi-graph GCN (ST-ED-RMGC) deep learning framework for short-term taxi OD prediction (*1*). The model utilizes a residual multi-graph GCN (RMGC) to explore various spatial relationships, employs an improved spatiotemporal LSTM to extract temporal features from OD demand, integrates temporal and spatial features through the encoder, and, finally, uses the decoder to reuse multiple RMGC networks to convert the compressed vector back into OD graphs for predicting target OD demands.

To address the issue of information overload caused by OD demand sequences, attention mechanisms are incorporated into the encoder-decoder framework (*33*). Various attention mechanisms, such as residual attention, multilayer attention, and spatial attention, have been proposed (*34–36*). Sankar et al. drew inspiration from Vaswani et al. and jointly learned the structure and temporal dynamics of directed graphs using attention mechanisms (*33*, *37*). Zhang et al. proposed a framework called a channel-wise attentive split-CNN (*38*). Zhang et al. proposed a spatiotemporal attention fusion network based on Chen et al. for short-term passenger flow prediction of urban rail transit systems during the New Year's Day period (*36*, *39*).

Problem Description

OD Diagram

Previous studies have often subdivided the spatial extent of the study region into regular grids, including squares and hexagons. This approach simplifies the training of machine learning algorithms, including CNN. However, this approach lacks a certain degree of interpretability and falls short of adequately capturing the administrative and functional characteristics of the regions under consideration. In this study, a delineation map of the Manhattan area in the U.S. was acquired, and this map subdivides the Manhattan area of New York City into multiple irregular regions according to administrative postal codes, as shown in Figure 1*a*. This dataset was acquired from government sources and is characterized by a high level of reliability. Concurrently, the day is partitioned into uniform time intervals. This study aims to forecast the order request count for all OD pairs within each time interval.

Moreover, acknowledging that distinct OD pairs may not have a direct geographical link, they may still manifest similarities in specific attributes, such as similar origins or destinations. In consideration of this, a customized OD graph $G = (V, E)$ is introduced in this paper, as illustrated in Figure 1*a*. In this context, V represents the graph network's vertices, each corresponding to an OD pair. E denotes the collection of edges, providing the flexibility to customize their weights to configure the desired spatial topological relationships, as shown in Figure 1*b*.

Research Problem and Features

Let $x_i^{(d,t)}$ denote the OD demand in the ith OD pair at the time interval t of day d, where $i \in V$ (the set of OD pairs). Let $X^{(d,t)}$ denote the OD demand in all OD pairs at the time interval t of day d. To predict the OD demand in all OD pairs at the time interval t of day d, this study leverages the historical OD demand data preceding the time interval t of day d to extract temporal and spatial features. However, because of limitations in computational resources, utilizing all historical OD data is neither feasible nor necessary. In short-term OD prediction, two primary types of temporal dependencies are typically present in future OD demand: trend-based, where the predicted OD demand is influenced by historical OD demand from several preceding intervals, and period-based, where the predicted OD demand is influenced by OD demand from the same time intervals in previous days or weeks (*40*). Considering these factors, this paper aims to extract the following data for feature mining:

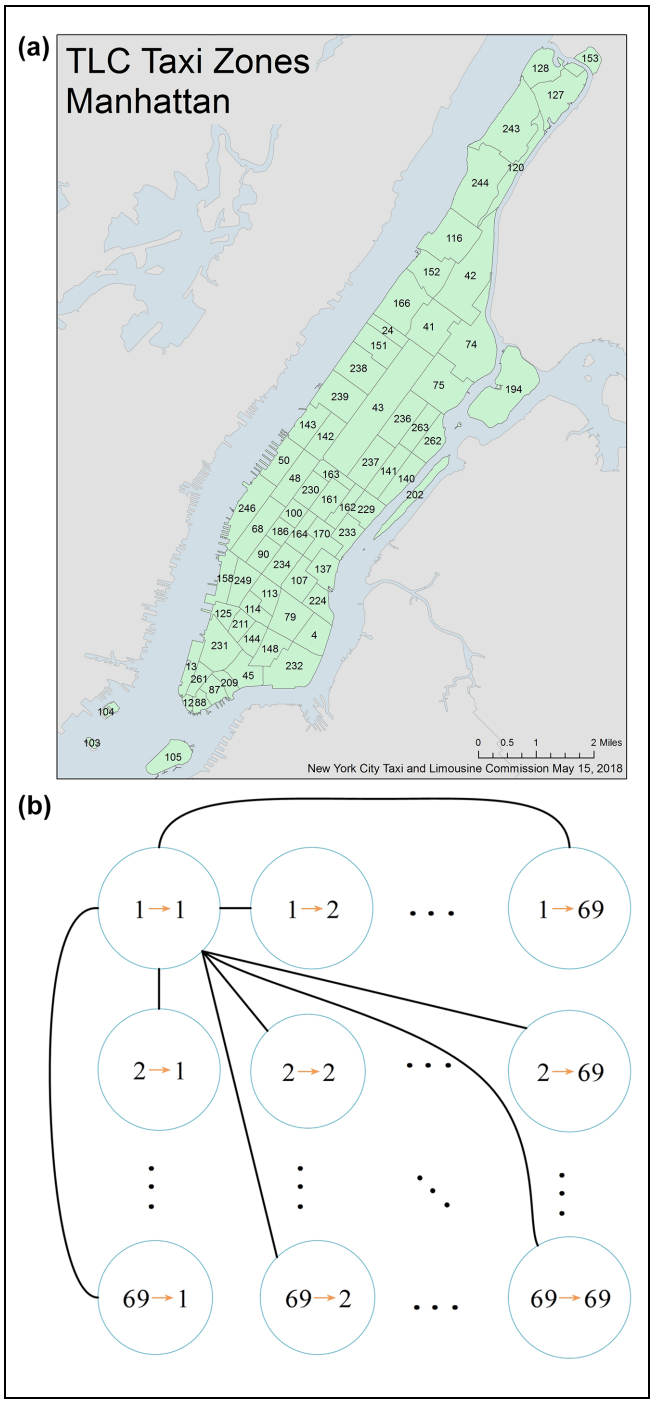

Figure 1. Regional division and custom network: (*a*) Regional division and (*b*) Custom network.

1) Trend-based features: the OD demand from the three preceding intervals of the predicted OD demand, that is, $X^{(d,t-1)}$, $X^{(d,t-2)}$, $X^{(d,t-3)}$.
2) Period-based features: the OD demand for the same time intervals on the first three calendar days preceding the target date of prediction OD demand, that is, $X^{(d-1,t)}$, $X^{(d-2,t)}$, $X^{(d-3,t)}$.

Then, the OD ride-sourcing demand prediction problem can be shown in Equation 1.

Problem 1: To learn a Function $f(\cdot)$: the next moment's OD demand is computed by applying $f(\cdot)$ to the historical demand of OD flows.

$$\tilde{X}^{(d,t)} = f(X^{(d,t-1)}, X^{(d,t-2)}, X^{(d,t-3)}, X^{(d-1,t)}, X^{(d-2,t)}, X^{(d-3,t)}) \quad (1)$$

Methods

Model Overview

Figure 2 illustrates the overview of the proposed RF-STED model in this study, which utilizes a spatial and temporal feature extraction-encoder-decoder framework. The model consists of four parts: the space module, time module, encoding module, and decoding module. The main functions of each module are as follows:

1) The spatial module translates the relationships between OD pairs into a spatial topology graph, extracting multiple spatial correlations among OD pairs through MGCN.
2) The time module reveals temporal correlations in various OD data patterns through MConv-LSTM, all while accounting for the inherent spatial characteristics of the OD data.
3) The encoding module captures spatiotemporal dependencies and converts them into a dense vector space.
4) The decoding module decompresses the compressed vectors, decodes them into the OD graph, and forecasts future OD demands through a residual feature extractor.

In the following sections, we will provide a detailed introduction to the composition of each component and their interrelationships.

Model Introduction

Spatial Dependence of OD Pairs. In the field of short-term taxi OD flow prediction research, the extraction of both OD pairs and spatial features within various predefined regions is of utmost significance. Certain OD pairs show similarities, such as those with stable passenger flows throughout the day. Similarly, certain regions might possess strong interconnections, such as between stations and city centers. These characteristics embody prominent spatial features, and the effectiveness of spatial-based

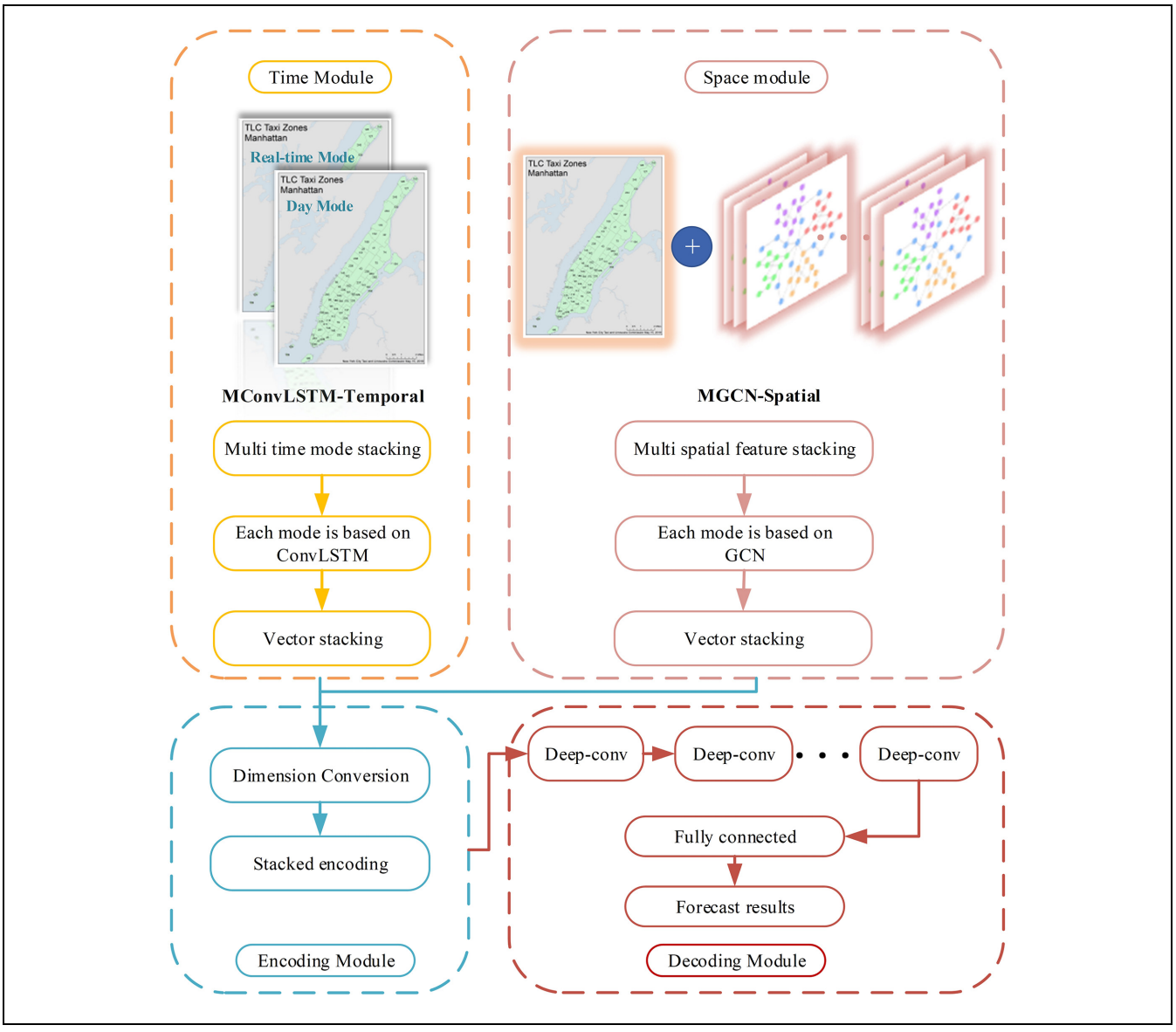

Figure 2. Spatiotemporal encoder-decoder network with residual feature extractor (RF-STED) architecture.
Note: Conv-LSTM = convolutional long short-term memory network; GCN = graph convolutional network; MConv-LSTM = multiple convolutional long short-term memory network; MGCN = multiple graph convolutional network.

OD data processing significantly influences the accuracy of short-term OD flow prediction.

In this study, we capture the relationships among OD demand using diverse custom graph structures. These relationships are examined from two perspectives—physical spatial relationships and logical spatial relationships—to encompass the spatial interactions among OD pairs. These relationships are represented through adjacency matrices denoted as A_i. This section primarily extracts the following four relationship types:

1) Starting and ending point relationship graph $G_n(V, E, A_n),\ A_n \in \mathbb{R}^{N \times N}$
2) Centroid graph $G_d(V, E, A_d), A_d \in \mathbb{R}^{N \times N}$
3) OD volume trend graph $G_g(V, E, A_g), A_g \in \mathbb{R}^{N \times N}$
4) Land property relationship graph $G_f(V, E, A_f)$, $A_f \in \mathbb{R}^{N \times N}$

G_n and G_d are part of physically oriented relationship graphs, whereas G_g and G_f are associated with logically oriented relationship graphs. Consequently, this study comprehensively explores the spatial topological relationships among OD pairs, incorporating both physical and logical dimensions.

Starting and Ending Point Relationship Graph. When the starting point or destination of two OD pairs is the same, a significant spatial relationship between these two OD

pairs is inevitably present. Therefore, the adjacency matrix is defined as shown in Equations 2–4.

$$[A_n^o]_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}, \forall i,j \in V \tag{2}$$

$$[A_n^D]_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}, \forall i,j \in V \tag{3}$$

$$[A_n]_{i,j} = \varepsilon_n^O [A_n^o]_{i,j} + \varepsilon_n^D [A_n^D]_{i,j}, \forall i,j \in V \tag{4}$$

where

$[A_n^o]$ = whether the starting points of any two OD pairs are the same,

$[A_n^D]$ = whether the destinations of any two OD pairs are the same,

$A_n \in \mathbb{R}^{N \times N}$ = the weighted average of the two adjacency matrices,

ε_n^O = the proportion of the origin, and

ε_n^D = the proportion of the destination.

Centroid Graph. When the starting points or destinations of two OD pairs are very close, these OD pairs also exhibit a certain degree of spatial relationship. Therefore, the adjacency matrix is defined as shown in Equations 5–7.

$$[A_d^o]_{i,j} = \frac{1}{\sqrt{(lng_i^O - lng_j^O)^2 + (lat_i^O - lat_j^O)^2}}, \forall i,j \in V \tag{5}$$

$$[A_d^D]_{i,j} = \frac{1}{\sqrt{(lng_i^D - lng_j^D)^2 + (lat_i^D - lat_j^D)^2}}, \forall i,j \in V \tag{6}$$

$$[A_d]_{i,j} = \varepsilon_d^O [A_d^o]_{i,j} + \varepsilon_d^D [A_d^D]_{i,j}, \forall i,j \in V \tag{7}$$

where

$[A_d^o]$ = the distance relationship between the starting points of two OD pairs,

$[A_d^D]$ = the distance relationship between the destinations of two OD pairs,

$A_d \in \mathbb{R}^{N \times N}$ = the weighted average of the two adjacency matrices,

ε_d^O = the proportion of the origin, and

ε_d^D = the proportion of the destination.

OD Volume Trend Graph: OD Pair Clustering. In some cases, the number of OD pairs can be extensive, yet different OD pairs may exhibit similar patterns of variation. Therefore, clustering OD pairs is crucial. This study considers clustering different OD pairs based on historical OD data. Previous research has indicated that the K-means clustering method is straightforward, easy to understand, and converges quickly, making it widely used (*41*). Similarly, this study adopts the K-means clustering method. The K-means clustering algorithm is a popular unsupervised clustering technique that iteratively minimizes the distance between cluster centroids and data points. Let X be an unlabeled dataset containing n samples, $X = [x(1), x(2), ..., x(n)]$, and let k be the number of clusters. The model aims to partition the data into k clusters, denoted as $C = [C_1, C_2, ..., C_k]$. The centroids of cluster C_i are defined by equation, and the objective function to minimize is shown in Equations 8–9.

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{8}$$

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{9}$$

The K-means algorithm, as shown in Figure 3*a*, commences by selecting k random points as the initial centroids for each cluster. Subsequently, the distances between all samples and the centroids are computed. Each sample is assigned to the cluster with the nearest centroid based on the computed distances. Then, the centroids are recalculated using the existing samples within each cluster. This process iterates to explore all feasible cluster assignments, aiming to derive the optimal solution.

In this study, four distinct features of OD pairs are employed for clustering: the average daily OD value, the variance of the daily OD values, the average OD value during the morning peak hours, and the average OD value during the evening peak hours. These features for the i th OD pair on day d can be computed using Equations 10–13. Initially, on the d_{th} day, data for all OD pairs are extracted as samples denoted by Z_d, as defined in Equation 14. Subsequently, the set Z is selected as the training data, as defined in Equation 15.

$$QAV_i^d = \frac{1}{N_a} \sum_t x_i^{(d,t)} \tag{10}$$

$$QVA_i^d = \frac{1}{N_a} \sum_t (x_i^{(d,t)} - QAV_i^d)^2 \tag{11}$$

$$QMP_i^d = \frac{1}{N_m} \sum_t x_i^{(d,t)} \tag{12}$$

$$QEP_i^d = \frac{1}{N_e} \sum_t x_i^{(d,t)} \tag{13}$$

$$Z_d = \begin{pmatrix} QAV_1^d & QVA_1^d & QMP_1^d & QEP_1^d \\ QAV_2^d & QVA_2^d & QMP_2^d & QEP_2^d \\ \cdots & \cdots & \cdots & \cdots \\ QAV_N^d & QVA_N^d & QMP_N^d & QEP_N^d \end{pmatrix} \tag{14}$$

$$Z = \frac{1}{D} \sum_d Z_d \tag{15}$$

where

QAV = the average daily OD value,

QVA = the variance of the daily OD values,

QMP = the average OD value during the morning peak hours,

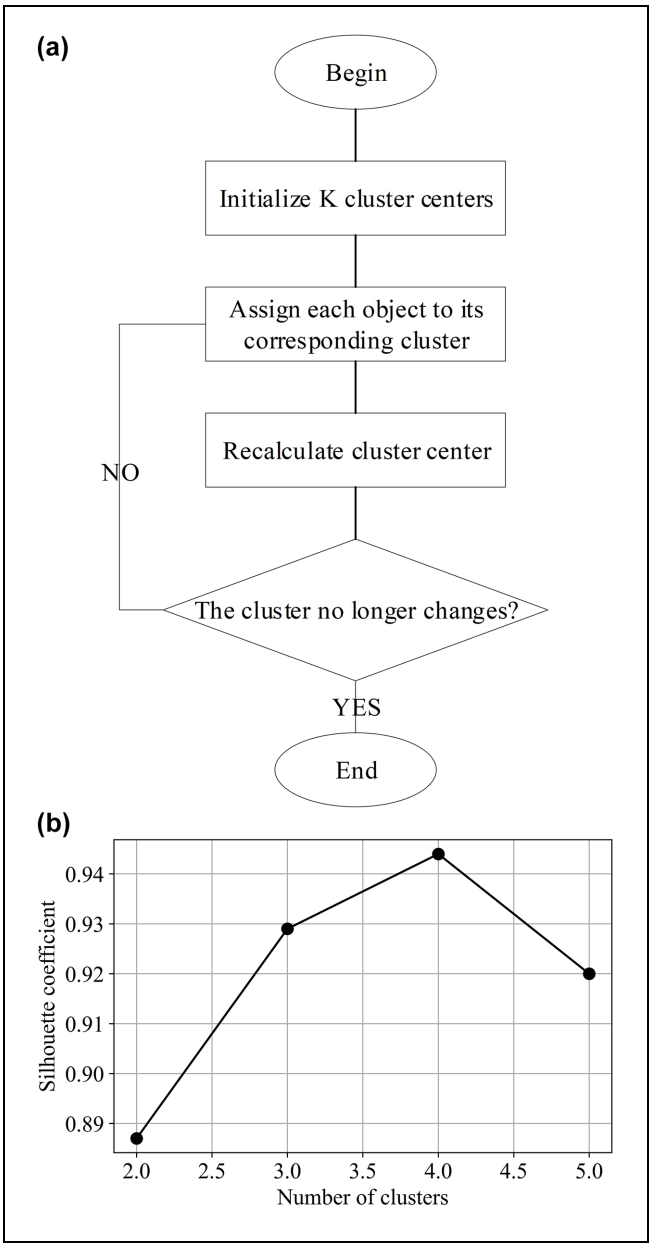

Figure 3. Algorithm procedure and results presentation.

QEP = the average OD value during the evening peak hours,

N_a = the number of time intervals divided on day d,

N_m = the number of time intervals divided by the day's morning peak, and

N_e = the number of time intervals divided by the day's evening peak.

To determine the optimal number of clusters, this paper sets the number of cluster centers to 2, 3, 4, and 5. The impact of different cluster numbers on clustering effectiveness was evaluated by the silhouette coefficient. For the K-means clustering algorithm, the silhouette coefficient ranges from -1 to 1. For a specific data point, a value closer to 1 indicates higher similarity with other samples in the same cluster and dissimilarity with samples in other clusters; conversely, a value closer to -1 implies the opposite. The clustering results are shown in Figure 3*b*. It can be observed from the figure that the clustering effectiveness is optimal when the number of clusters is set to four. Therefore, the classification process starts with setting the number of cluster centers to four.

Subsequently, the clustering results are subjected to visualization. For ease of interpretation, this study groups the average values of morning peak OD and evening peak OD, along with their corresponding cluster assignments, into the first group. The results are shown in Figure 4*a*. The second group is formed by considering the average daily OD value and variance, and its presentation is shown in Figure 4*b*. Figure 4*a* illustrates noticeable distinctions in morning and evening peak flow among the four clusters of OD pairs. Figure 4*b* illustrates that OD pairs with larger morning and evening peak flows tend to exhibit greater average daily OD values and variances.

OD Volume Trend Graph: Definition of the Relationship Graph. After the clustering process, this study has categorized all OD pairs into four distinct clusters. OD pairs within the same cluster exhibit substantial logical spatial relationships. Therefore, the adjacency matrix is defined as shown in Equation 16.

$$[A_g^o]_{i,j} = \begin{cases} 1 & i \text{ and } j \text{ are the same category} \\ 0 & \text{otherwise} \end{cases}, \forall i,j \in V \tag{16}$$

Land Property Relationship Graph. When the land use characteristics of two OD pairs are relatively similar (such as both OD pairs having their origins at shopping malls or both OD pairs having their destinations at subway stations) it can be assumed that there exists a substantial logical spatial correlation between these two OD pairs. Therefore, the adjacency matrix is defined as shown in Equations 17–19.

$$[A_f^O]_{i,j} = \frac{1}{\sqrt{\sum_k (x_{k,j}^O - x_{k,j}^O)^2}}, \forall i,j \in V \tag{17}$$

$$[A_f^D]_{i,j} = \frac{1}{\sqrt{\sum_k (x_{k,j}^D - x_{k,j}^D)^2}}, \forall i,j \in V \tag{18}$$

$$[A_f]_{i,j} = \varepsilon_f^O [A_f^o]_{i,j} + \varepsilon_f^D [A_f^D]_{i,j}, \forall i,j \in V \tag{19}$$

where

$[A_f^o]$ = the land use characteristic association of the OD pair's origin,

$[A_f^D]$ = the land use characteristic association of the OD pair's destination,

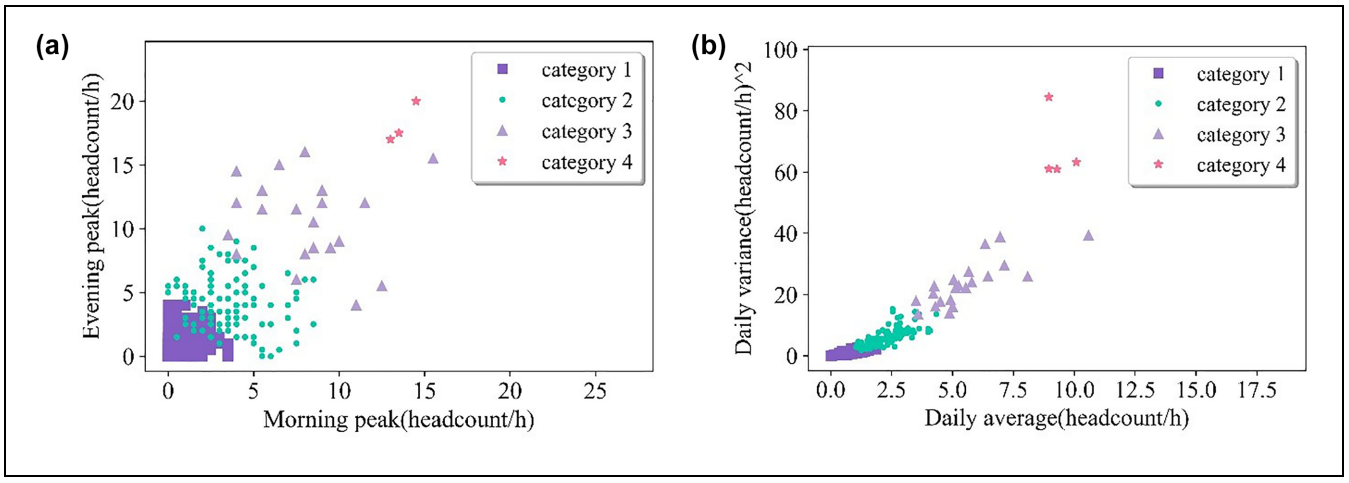

Figure 4. Clustering effect: (*a*) Morning Peak - Evening Peak and (*b*) Mean-Variance.

$A_f \in \mathbb{R}^{N \times N}$ = the weighted average of the two critical matrices,

ε_f^O = the proportion of the origin, and

ε_f^D = the proportion of the destination.

For each region's land use characteristics, this study adopts measurement criteria encompassing metrics highly correlated with land usage types and travel patterns, including housing density, population density, employment density, road density, and average distance to the nearest transportation facility, among others.

Multi-Graph Convolutional Network (MGCN). In this subsection, we introduce MGCN, which consists of multiple GCN networks, and serves as the primary tool for spatial feature extraction. Now, let us delve into its fundamental component: GCN.

GCN has gained popularity in recent years as a neural network architecture (*42*). As the field has evolved, researchers in the domain of graph networks have proposed various architectures, with GCN holding a prominent position. In the context of traffic demand prediction, GCN has exhibited promising predictive capabilities. This is attributed to its ability to enable researchers to define relationships between nodes, allowing machines to intelligently learn such relationships. In transportation research, the ability to customize spatial relationship patterns is of paramount importance. For instance, in spatial terms, two points might be physically close but not directly connected, requiring researchers to intervene to nullify their spatial association. This scenario frequently arises in traffic prediction, explaining the widespread utilization of GCN in this domain.

Presently, graph-based convolutions can be broadly classified into two major categories. The first category involves spectral-based graph convolutions, which use Fourier transformation to map nodes into the frequency-domain space, perform convolutions in the frequency-domain space, and subsequently map the feature products back to the time-domain space (*43*, *44*). The second category revolves around spatial-domain graph convolutions, closely resembling traditional CNNs but with increased complexity because of the challenge of defining node neighbors and their relationships within the graph structure (*45*, *46*). In this paper, we adopt the first category, namely spectral-based graph convolutions, to build the foundational GCN model. The GCN convolution operation is defined in Equation 20.

$$H^{l+1} = f(H^l, A) = \sigma(D^{-1/2}A^*D^{-1/2}H^lW^l + b^l) \quad (20)$$

where

H^{l+1} = the output resulting from convolution with a user-defined spatial structure (containing learned features with the number of features specified by the model),

H^l = the input feature vector (encompassing the feature vectors of individual nodes),

σ = the activation function (commonly used functions include rectified linear unit [ReLU], linear, etc.),

Wl = the trainable weight matrix (connecting the lth and $(l+1)$th hidden layers),

$A^* = A + I$ = the self-connected adjacent matrix augmented by an identity matrix (employed to maintain self-information during convolution),

D = the degree matrix, and

b = the bias vector for the l^{th} layer.

To extract various spatial relationships among OD demand, we devise the training architecture as shown in Figure 5. Assuming J types of spatial relationships correspond to J adjacency matrices ($j \in \{1, \cdots, J\}$), the input vector for OD data, denoted as $\tilde{X}^{(d,t)} \in \mathbb{R}^{n \times n \times d \times t}$, where n represents the number of

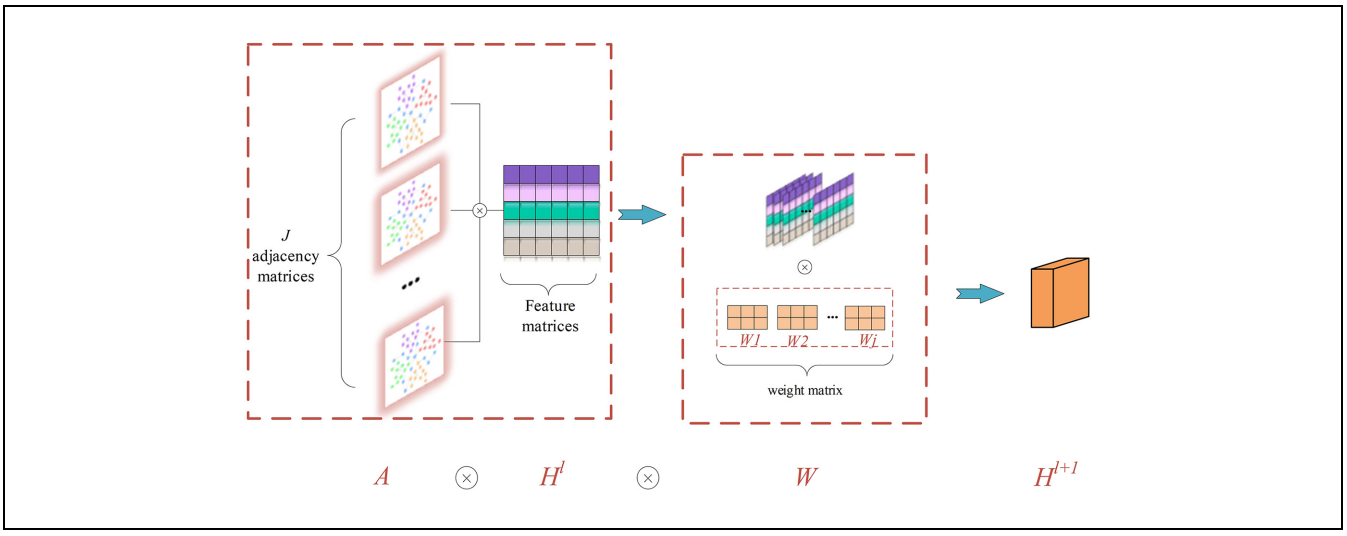

Figure 5. Multiple graph convolutional network (MGCN) architecture.

partitioned areas , d represents the number of days in advance of the prediction time, and t represents the time interval in advance of the prediction time. Initially, we merge the first two dimensions and the last two dimensions of the $\tilde{X}^{(d,t)}$ vector to obtain a new vector $\tilde{X}^{(d\times t)}\in\mathbb{R}^{N\times F}$, where N represents the number of OD pairs, F represents the number of features of OD data, incorporating data from preceding days and time intervals as features. Subsequently, $\tilde{X}^{(d\times t)}$, along with J adjacency matrices $A\ \in\mathbb{R}^{n\times n}$, serves as input for the MGCN network. Finally, the output vectors of each GCN network are flattened and stacked in corresponding positions, yielding the output vector $\tilde{X}^{MGCN}\in\mathbb{R}^{N\times O\times J}$.

Multi-Convolutional Long Short-Term Memory Network (MConv-LSTM). We now introduce the MConv-LSTM network, which utilizes the Conv-LSTM network as the foundational component to capture the temporal relationships among OD quantities. To capture the temporal relationships of OD quantities under varying temporal patterns, this study devises a multi-channel Conv-LSTM network to constitute the MConv-LSTM network, serving as the fundamental element for temporal feature extraction.

The Conv-LSTM network was initially introduced in a study related to precipitation prediction titled "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting" (*29*). Precipitation forecasting involves predicting future data based on past data, making it a temporal problem. Previous research mainly relied on LSTM networks for this purpose (*47*). However, LSTM networks had a limitation—their input vector dimension was two-dimensional (2D), where one dimension represented time, and the other dimension represented features. Precipitation data, distributed across spatial dimensions, required at least a three-dimensional (3D) input vector, with one dimension for time, one for features (precipitation amount), and an additional dimension for spatial locations. The paper highlighted that directly incorporating spatial information into the feature dimension, as in LSTM, would result in a loss of spatial features. To address this, the paper introduced a novel network structure called Conv-LSTM, which not only captured LSTM-like temporal relationships but also had spatial feature extraction capabilities similar to CNN.

Let us begin by understanding the architecture of LSTM. The basic definition of LSTM is given by Equations 21–25. The Conv-LSTM proposed in the paper replaces some connections with convolutional operations, as defined by Equations 26–30.

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{21}$$

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{22}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_c \times [h_{t-1}, x_t] + b_c) \tag{23}$$

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{24}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{25}$$

$$f_t = \sigma(W_f * [H_{t-1}, X_t] + b_f) \tag{26}$$

$$i_t = \sigma(W_i * [H_{t-1}, X_t] + b_i) \tag{27}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_c \times [H_{t-1}, X_t] + b_c) \tag{28}$$

$$o_t = \sigma(W_o * [H_{t-1}, X_t] + b_o) \tag{29}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{30}$$

where

f_t = the forget gate (determining how much of the previous state can be retained for the current time step),

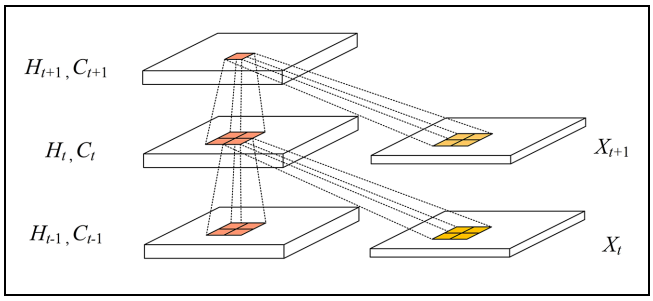

Figure 6. Convolutional long short-term memory (Conv-LSTM) network convolutional operation.

i_t = the input gate (deciding how much of the current network input can be stored in the cell state)
c_t = the cell state at time step t,
o_t = the output gate (controlling how much of the cell state can be output to the current value h_t),
x_t = the input vector at time step t,
W = trainable weights,
b = the bias vector,
σ = the activation function,
X_t = the vector on the spatial network, and
H_t = the output at time step t on the spatial network (indicated by the convolution operation "*" as shown in Figure 6).

To capture temporal relationships among OD demand with multiple patterns, we have devised the training framework, as shown in Figure 7. OD data resides in a 2D space, and when accounting for the temporal aspect, a third dimension (time dimension) is introduced to accommodate the input vector dimension of Conv-LSTM. Additionally, Conv-LSTM not only captures the temporal characteristics of OD demand through time sequences but also extracts spatial features from the OD data.

This paper extracts two time patterns: the input vector for OD data is represented as $\tilde{X}^{(d\times t)} \in \mathbb{R}^{n\times n\times d\times t}$, where n represents the number of partitioned regions, t represents OD time series at trend-based features

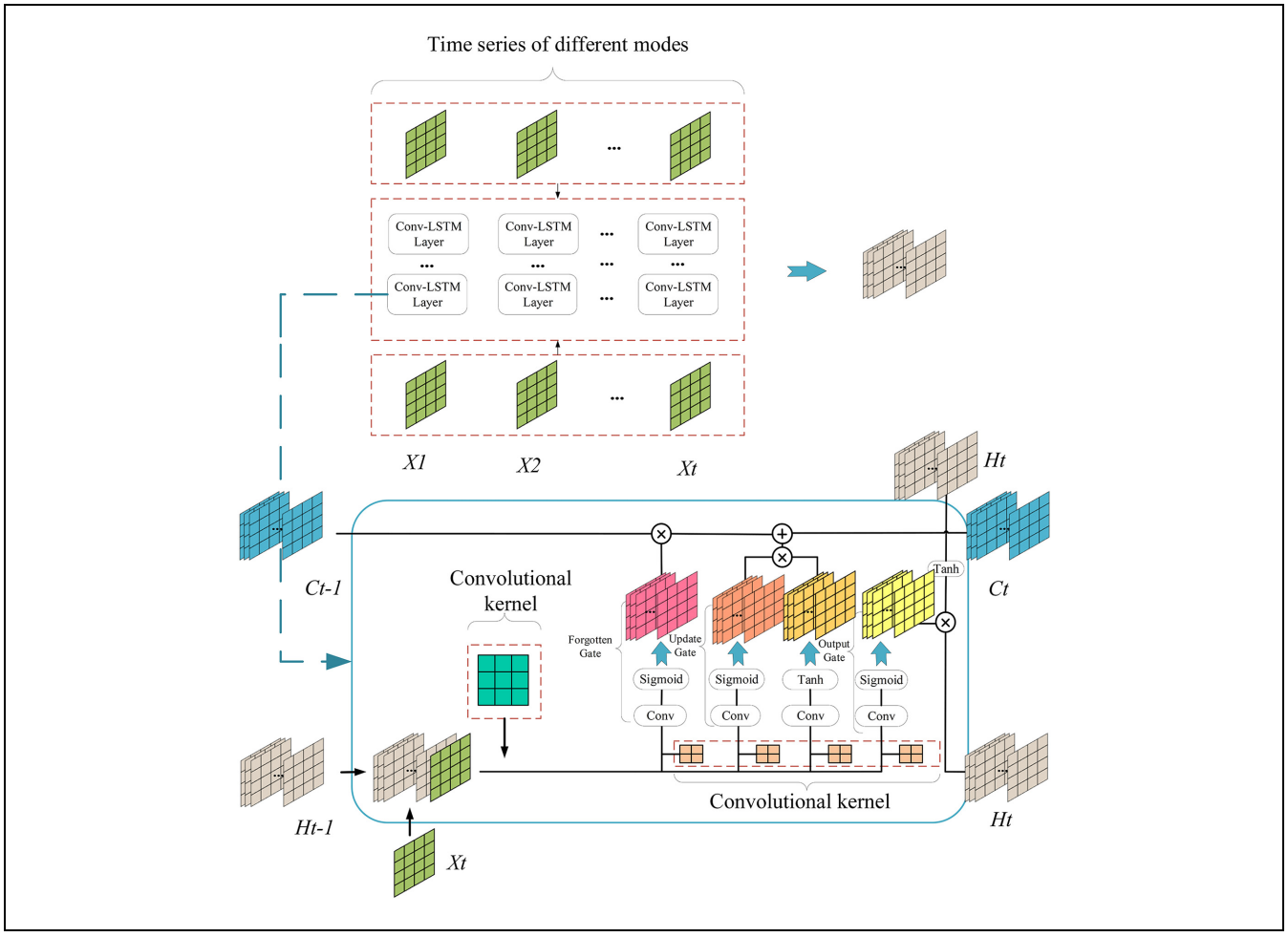

Figure 7. Multiple convolutional long short-term memory network (MConv-LSTM) architecture.
Note: Conv-LSTM = convolutional long short-term memory network.

pattern, and d represents OD time series at period-based features pattern. Extract two types of time pattern data from $\tilde{X}^{(d\times t)}$ as input vectors for Conv-LSTM, namely $\tilde{X}^{t} \in \mathbb{R}^{n\times n\times t}$ and $\tilde{X}^{d} \in \mathbb{R}^{n\times n\times d}$. Subsequently, the output of each Conv-LSTM network is flattened and stacked at corresponding positions to form the output vector $\tilde{X}^{MConvLSTM} \in \mathbb{R}^{n\times n\times O\times I}$.

Encode Layer and Residual Feature Extractor. Based on the above introduction, we first convert the first dimension of the output vector of the MGCN network into two dimensions with original region identifiers, while merging the last two dimensions to obtain a new vector, $\tilde{X}^{\mathrm{MGCN}'} \in \mathbb{R}^{n\times n\times (O*J)}$. Similarly, we combine the output vector of the MConv-LSTM network in the last two dimensions to obtain a new vector, $\tilde{X}^{\mathrm{MConLSTM}'} \in \mathbb{R}^{n\times n\times (O*I)}$. Finally, we concatenate and sum them along the third dimension to encode and obtain the output vector of the encoding layer, $\tilde{X}^{\mathrm{Encoder}} \in \mathbb{R}^{n\times n\times 1}$, as shown in Equation 31.

$$\tilde{X}^{\mathrm{Encoder}} = \mathrm{sum}\left(\mathrm{cat}\left(\mathrm{reshape}\left(\tilde{X}^{\mathrm{MGCN}}\right), \mathrm{reshape}\left(\tilde{X}^{\mathrm{MConvLSTM}}\right)\right)\right) \tag{31}$$

Now, let us introduce the residual feature extractor, a fundamental component of our proposed model. Its primary role is to decode the output vector generated by the encode layer and predict the required OD data. The key element of the residual feature extractor is the deep convolution (Deep-Conv) network. Each Deep-Conv network utilizes convolutional kernels of varying sizes to extract diverse features from the encoded data. Its fundamental definition is shown in Equation 32. The convolution operation entails applying the kernel to the input feature map via a sliding window, multiplying the corresponding elements in the window by their counterparts in the convolution kernel, and subsequently summing them to obtain the values at positions (i, j) on the output feature map. Such convolution operations are proficient at capturing localized features within the input data.

This paper continues to use the output of one Deep-Conv network as the input of the next Deep-conv network, as shown in Figure 8. Each Deep-Conv network utilizes distinct feature extractors, and we introduce residual networks to address the challenges of gradient vanishing and explosion, which are common issues in deep network architectures.

The residual network was originally proposed by four scholars from Microsoft Research (*48*). Simply put, for a layer of a neural network, the input it can receive comes from the output of its previous layer of a network, but it cannot perceive the output of the previous layer of the network. If the previous layer of the network does not improve the accuracy of the model's prediction, or even leads to a decrease in the accuracy of the model, it cannot obtain this information. The idea of a residual network is to move it up one layer The output of the previous layer is also used as its own input so that it can sense the output of the previous layer. If the previous network does not improve the model performance, it can choose to omit or weaken, thereby improving the flexibility of the model. Its basic definition is as shown in Equation 33. Therefore, the residual feature extractor is defined as Equation 34.

The output vector of the encoding layer $\tilde{X}^{\mathrm{Encoder}}$ contains both the spatial and temporal features of the historical OD demand. To further extract spatiotemporal features, we utilize $\tilde{X}^{\mathrm{Encoder}}$ as the input vector for the residual feature extractor. We can finally obtain the estimated demand of all OD pairs, that is, $\hat{X}^{(d,t)}$. Formally, the decoder architecture can be defined as Equation 35.

$$S_{l+1} = \left(\prod_{i=2}^{I}\sum_{m}\sum_{n} K_i(m,n)\right)\left(\sum_{m}\sum_{n} K_i(m,n) S_l(i-m, j-n)\right) \tag{32}$$

$$x_{l+1} = f(h(x_l) + \gamma(x_l, W_l)) \tag{33}$$

$$RS_{l+1} = f\left(h(RS_l) + \sum_{k=2}^{K} \mathrm{Dep} - \mathrm{cov}(RS_l^{k-1}, \mathrm{cov}(k,k))\right) \tag{34}$$

$$\hat{X}^{(d,t)} = f\left(h\left(\tilde{X}^{\mathrm{Encoder}}(k)\right) + \sum_{k=\mathrm{s}}^{K} \mathrm{Dep} - \mathrm{cov}(\tilde{X}^{\mathrm{Encoder}}(k-1), \mathrm{cov}(k,k))\right) \tag{35}$$

where

S_{l+1} = the output vector of the Deep-Conv network,
K = the convolutional kernel,
S_l = the input vector to the Deep-Conv network,
m and n = the indices for the convolutional kernel,
x_l = the input vector of the layer network,
x_{l+1} = the output vector of the layer network,
γ = a residual function,
$h(\cdot)$ = a function to complete the mapping from x_l to x_{l+1} dimensions,
$f(\cdot)$ = the activation function,
s = the initial convolution kernel size, and
K = the convolution kernel size of the last Deep-Conv network.

Let W, b be all the trainable weights and biases in the whole encoder-decoder architecture, we can train the weights and biases by solving the following optimization problem:

$$\min_{W,b} \sum_{d}\sum_{t} \left\| \hat{X}^{(d,t)} - \tilde{X}^{(d,t)} \right\|_2^2 + \alpha \|W\|_2^2 \tag{36}$$

The initial term in the expression aims to minimize the squared loss between the predicted OD demand pattern

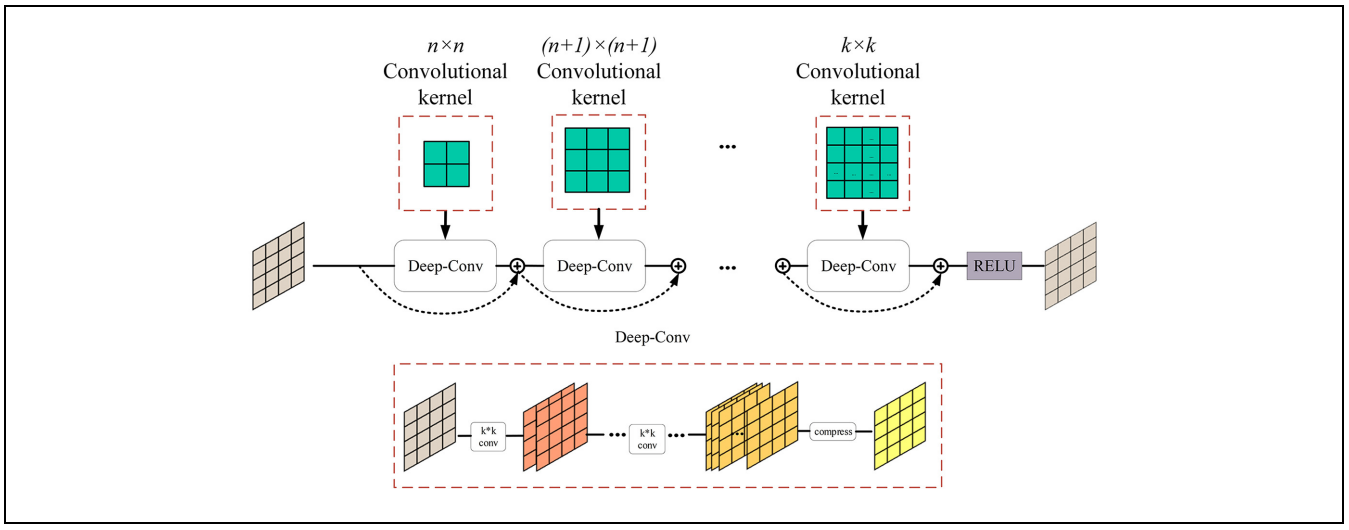

Figure 8. Residual feature extractor architecture.
Note: Deep-Conv = deep convoltion network; RELU = rectified linear unit.

and the actual pattern, while the subsequent term introduces an L2-norm regularization component to prevent the emergence of overly intricate models that may result in overfitting. The training algorithm of the model is presented in (Algorithm 1).

Evaluation

This section comprehensively compares and analyzes the proposed model from several perspectives, including overall predictive performance, performance under different types of OD pairs, and robustness analysis through ablation experiments. The comparison includes model prediction accuracy and robustness.

Dataset Introduction

The data for this study originates from the Taxi and Limousine Commission website, which provides data on New York City's taxis and shared bikes, and related information. In this research, the model's evaluation and validation are conducted using processed taxi data from the Manhattan area covering the period from January 1, 2022, to January 31, 2022. The dataset includes fields such as "Data Sequence Number," "Trip Start Time," "Trip End Time," "Trip Start Zone ID," and "Trip End Zone ID."

Because of recording tools or other factors, certain data entries might lack significance or contain noticeable errors. Therefore, certain criteria are applied to filter out unreliable data, such as removing entries with start times later than end times, entries with time intervals greater than 2 h (as taxi trips are generally not excessively long), entries with time intervals less than 5 min (considered as invalid because of extremely short durations), and entries with missing fields. After data cleansing, the resulting dataset comprises 2,463,930 valid records.

Subsequently, these valid records are transformed into OD data. The day is divided into 24 intervals, leading to a multidimensional OD data structure of dimensions 31 (days) * 24 (time intervals) * 69 (number of zones) * 69 (number of zones).

Model Configuration

Configuration of the Proposed Model. The proposed RF-STED model is implemented using PyTorch. Following common practices in previous research, a standard train-test split is adopted with 70% of the data used for training and 30% for testing. As described earlier, the data is structured into 24 time intervals per day over a continuous span of 31 days. Therefore, this study employs the first 22 days of data for training and the remaining 9 days for testing.

In the context of the MConv-LSTM network, two types of time feature are extracted: trend-based and cyclic-based time features. The time steps are set to 3. Consequently, two Conv-LSTM networks are parallelly employed within the MConv-LSTM component. Each Conv-LSTM network is equipped with 3×3 convolutional kernels, a single hidden layer, 1 input feature, and 32 output features, and employs the ReLU activation function.

In the context of the MGCN network, four spatial relationships are extracted: OD relation, centroid relation, OD flow trend relation, and OD land property relation. As a result, four GCN networks are concatenated within the MGCN component. The time features of the

Algorithm 1. ST-ED-RMGC training algorithm

Input OD pair number $i \in V$,
Historical demand of all OD pairs $\{\boldsymbol{X}^{(d,t-1)}, \boldsymbol{X}^{(d,t-2)}, \boldsymbol{X}^{(d,t-3)}, \boldsymbol{X}^{(d-1,t)}, \boldsymbol{X}^{(d-2,t)}, \boldsymbol{X}^{(d-3,t)}\}$,
The graphs $\boldsymbol{G}$: Starting and ending point relationship graph $G_n(V, E, \boldsymbol{A_n})$;
Centroid graph $G_d(V, E, \boldsymbol{A_d})$;
OD volume trend graph $G_g(V, E, \boldsymbol{A_g})$;
Land Property Relationship graph $G_f(V, E, \boldsymbol{A_f})$
Output RF-STED with well-trained parameters $\boldsymbol{W}$;
OD demand to be predicted $\boldsymbol{X}^{(d,t)}$
//Construct a set of input-output instances $\mathcal{H}$
Initialize a null set: $\boldsymbol{\mathcal{H}} \leftarrow \emptyset$
for day interval d ($4 \leqslant d \leqslant D$) **do**
for time interval t ($4 \leqslant t \leqslant T$) **do**
Get temporal features of all OD pairs at each time interval:
$\tilde{\boldsymbol{X}}^{(d,t)} = [\boldsymbol{X}^{(d,t-1)}, \boldsymbol{X}^{(d,t-2)}, \boldsymbol{X}^{(d,t-3)}, \boldsymbol{X}^{(d-1,t)}, \boldsymbol{X}^{(d-2,t)}, \boldsymbol{X}^{(d-3,t)}]$
Put training sample into the dataset: $\boldsymbol{\mathcal{H}} \leftarrow \boldsymbol{\mathcal{H}} + (\tilde{\boldsymbol{X}}^{(d,t)}, \boldsymbol{X}^{(d,t)})$
End for
Divide $\boldsymbol{\mathcal{H}}$ into training and test datasets $\boldsymbol{\mathcal{H}}_{\mathbf{train}}$, $\boldsymbol{\mathcal{H}}_{\mathbf{valid}}$, $\boldsymbol{\mathcal{H}}_{\mathbf{test}}$
//Training RF-STED model
Initialize the hidden status, all weights and bias parameters
Calculate the four adjacency matrices ($\boldsymbol{A_n}$, $\boldsymbol{A_d}$, $\boldsymbol{A_g}$, $\boldsymbol{A_f}$) according to Eq(4), Eq(7), Eq(16), Eq(19)
For $n = 1 \rightarrow$ number of epoch **do**
Randomly select a batch of sample $\boldsymbol{\mathcal{H}}_b$ from $\boldsymbol{\mathcal{H}}_{train}$, where b = 1,2, ..., B
For space pattern j ($1 \leqslant j \leqslant 4$) **do**
Reshape the $\tilde{\boldsymbol{X}}_b^{(d,t)}$ tensor to fit the GCN module: $\tilde{\boldsymbol{X}}_b^{(d\times t)} \leftarrow$ Reshape $(\tilde{\boldsymbol{X}}_b^{(d,t)})$
Obtain the output of GCN by passing GCN networks (Eq(20)):
$\tilde{\boldsymbol{X}}_b^{\mathrm{GCN}_j} \leftarrow \sigma(\boldsymbol{D}^{-1/2}\boldsymbol{A}_j^*\boldsymbol{D}^{-1/2}\tilde{\boldsymbol{X}}_b^{(d\times t)}\boldsymbol{W}^l + \boldsymbol{b}^l)$
Reshape the $\tilde{\boldsymbol{X}}_b^{\mathrm{GCN}_j}$ tensor to fit the Residual Feature Extractor module:
$\tilde{\boldsymbol{X}}_b^{\mathrm{GCN}_j'} \leftarrow$ Reshape $(\tilde{\boldsymbol{X}}_b^{\mathrm{GCN}_j})$
End for
For time pattern i ($1 \leqslant i \leqslant 2$) **do**
Select the corresponding time pattern data $\tilde{\boldsymbol{X}}_b^{\boldsymbol{t}}$ ($i = 1$) or $\tilde{\boldsymbol{X}}_b^{\boldsymbol{d}}$ ($i = 2$) from $\boldsymbol{\mathcal{H}}_b$:
Obtain the output of Conv-LSTM by passing Conv-LSTM networks (Eq(21)~Eq(30)): $\tilde{\boldsymbol{X}}_b^{\mathrm{ConvLSTM}_i} \leftarrow$ ConvLSTM $(\widetilde{\boldsymbol{X}_b^{\boldsymbol{t}}}$ ($i = 1$) or $\tilde{\boldsymbol{X}}_b^{\boldsymbol{d}}$ ($i = 2$))
Reshape the $\tilde{\boldsymbol{X}}_b^{\mathrm{ConvLSTM}_i}$ tensor to fit the Residual Feature Extractor module:
$\tilde{\boldsymbol{X}}_b^{\mathbf{ConLSTM}_i'} \leftarrow$ Reshape $(\tilde{\boldsymbol{X}}_b^{\mathbf{ConvLSTM}_i})$
End for
Obtain the output of Encoder by summing the output vectors of MGCN and MConvLSTM at the feature index position(Eq(31)):
$\tilde{\boldsymbol{X}}_b^{\mathbf{Encoder}} \leftarrow SUM\ (\tilde{\boldsymbol{X}}_b^{\mathbf{GCN}_j}, \tilde{\boldsymbol{X}}_b^{\mathbf{ConLSTM}_i'})$
Let $\tilde{\boldsymbol{X}}_b^{\mathbf{Decoder}}(\boldsymbol{k}) = \tilde{\boldsymbol{X}}_b^{\mathbf{Encoder}}$, where k = 30
For decoding convolution kernel size k ($31 \leqslant k \leqslant 33$) **do**
Obtain the output of Decoder passing Residual Feature Extractor with kernel size k (Eq(32–34)):
$\tilde{\boldsymbol{X}}_b^{\mathbf{Decoder}}(\boldsymbol{k}) \leftarrow \mathrm{RF}\ (\tilde{\boldsymbol{X}}_b^{\mathbf{Decoder}}(\boldsymbol{k}-1))$
End for
Estimate the demand by obtaining the output of the last Residual Feature Extractor:
$\hat{\boldsymbol{X}}^{(d,t)} \leftarrow \tilde{\boldsymbol{X}}_b^{\mathbf{Decoder}}(\boldsymbol{k})$
Optimize $\boldsymbol{W}$ by minimizing loss function Eq(36)
End for

OD demand are utilized as vertex features in each OD pair. Consequently, the input feature dimension is set to six, the output feature dimension is 32, and the ReLU activation function is employed.

In the context of the residual feature extractor, we selected multi-layer Deep-Conv networks to perceive encoding features and decode them. Each Deep-Conv network has an input channel of 1 and an output

channel of 1. The convolutional kernel size of the DeepConv network ranges from 2×2 starts to increase linearly. Finally, set a fully connected layer with the ReLU activation function as the activation function.

The architecture and hyperparameters of each network component are carefully designed to ensure optimal feature extraction and combination, enhancing the overall predictive capacity of the RF-STED model. These choices are based on established research practices and empirical experience, providing a robust foundation for the model's performance in predicting short-term OD flows.

Configuration of Baseline Models. A total of 10 baseline models and the proposed RF-STED model are compared and analyzed in this study. The feather of each model and their architectural details in the context of this study are described below:

1) **ARIMA** is a classical time series analysis method commonly used for predicting future trends and patterns in time series data (*5*). It combines autoregressive and moving average concepts along with integration operations. The model's parameters are customized for different OD pairs.
2) **2D CNN** has been widely used in the field of artificial intelligence, particularly in image processing (*49*). It can simulate the time series of OD data using input channels. The internal convolutional kernels extract spatial and temporal features between regions and time series characteristics of OD data. In this study, the convolutional kernel size is set to 2×2, input channels are 6, and output channels are 1. A fully connected layer with a ReLU activation function is included.
3) **3D CNN** is an extension of CNN for processing 3D data such as volume data, video data, and time series data (*50*). It is particularly suitable for handling data with temporal dimensions. The model captures both spatial and temporal features effectively. The convolutional kernel size is set to $2 \times 2 \times 2$, input and output channels are one, and a fully connected layer with ReLU activation function is included.
4) **LSTM** is a variant of the recurrent neural network (RNN) designed to handle time series data (*21*). It addresses long sequences and dependencies better than standard RNNs, mitigating the vanishing gradient problem. A single LSTM layer with two hidden neurons is included, followed by a fully connected layer with ReLU activation function.
5) **Gated recurrent unit (GRU)** is another variant of RNNs designed for time series data (*51*). It has fewer parameters compared with LSTM, reducing the risk of overfitting and improving training speed. A single GRU layer with two hidden neurons is included, followed by a fully connected layer with ReLU activation function.
6) **GCN** extends the concept of CNN to topological graphs, capturing topological features between vertices (*52*). It captures the relationships and influences between different locations in OD data, enhancing prediction accuracy. The convolutional kernel size is set to 3×3, and a fully connected layer with ReLU activation function is included.
7) **Conv-LSTM** combines the features of CNN and LSTM, handling spatiotemporal sequence data (*29*). It captures spatial features between regions and effectively extracts temporal patterns from OD time series data. A single Conv-LSTM layer with two hidden neurons, a 3×3 convolutional kernel, and a fully connected layer with ReLU activation function are included.
8) **Traffic GCN (T-GCN)** combines GCN and GRU, extracting spatial features using GCN and then passing them through GRU for temporal processing (*23*). It is effective in handling traffic graph data with temporal sequences. A GCN layer with a 3×3 convolutional kernel is followed by a GRU layer with two hidden neurons and a fully connected layer with ReLU activation function.
9) **Temporal residual network (T-ResNet)** combines one-dimensional (1D) CNN and residual networks for time series data, particularly suited for data with temporal dependencies (*53*). It captures dynamic patterns and trends in time series data. The model includes two residual blocks, 1D convolutional layers with a kernel size of 3, and a fully connected layer with ReLU activation function.
10) **Spatio-temporal residual network (ST-ResNet)** combines 2D CNN and residual networks for data with both spatial and temporal relationships (*40*). It effectively captures both types of information, enhancing prediction accuracy. The model includes two residual blocks, 2D convolutional layers with a kernel size of 3×3, and a fully connected layer with ReLU activation function.

Model Comparison

To evaluate the performance of the model, mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are utilized as evaluation metrics.

Their basic expressions are given by Equations 37–39, respectively.

$$\mathrm{MSE} = \frac{1}{MN}\sum_{o=1}^{M}\sum_{d=1}^{N}\left(h_{i,o,d}^{t} - \hat{h}_{i,o,d}^{t}\right)^2 \quad (37)$$

$$\mathrm{MAE} = \frac{1}{MN}\sum_{o=1}^{M}\sum_{d=1}^{N}\left|h_{i,o,d}^{t} - \hat{h}_{i,o,d}^{t}\right| \quad (38)$$

$$\mathrm{MAPE} = \frac{1}{MN}\sum_{o=1}^{M}\sum_{d=1}^{N}\left|\frac{h_{i,o,d}^{t} - \hat{h}_{i,o,d}^{t}}{h_{i,o,d}^{t} + 1}\right| \quad (39)$$

where

$h_{i,o,d}^{t}$ = the actual OD value from origin region o to destination region d at the ith day and tth time interval, and

$\hat{h}_{i,o,d}^{t}$ = the predicted OD value for the same conditions.

The predictive performances of the RF-STED model and all benchmark models are shown in Table 1. The results in the table demonstrate that models combining spatial and temporal features, including the Conv-LSTM model and T-GCN, outperform deep learning models that focus solely on either temporal features (e.g., LSTM, GRU) or spatial features (e.g., 2D CNN, GCN). In contrast, traditional forecasting models, such as ARIMA, exhibit limited predictive capacity. They encounter challenges in handling complex spatial-temporal relationships, necessitating separate modeling for each OD pair, resulting in increased complexity in the prediction process. The RF-STED model proposed in this study demonstrates superior performance by enhancing the extraction process of both spatial and temporal features, consequently improving predictive accuracy. The essential logical relationships among OD pairs, crucial for short-term OD flow prediction, should not be disregarded. This experiment, to a certain extent, highlights the superiority of deep learning models over traditional approaches in short-term OD flow prediction. Moreover, the collaborative combination of fundamental deep learning models outperforms individual models.

Forecast Results

In this study, we randomly selected two specific time points and generated 3D visualizations to depict the distribution of actual and predicted OD demand values. Figure 9, *a* and *c*, illustrates the actual OD demand values and Figure 9, *b* and *d*, illustrates the predicted OD demand values. An examination of the figures reveals that the distribution of OD demand in this area is non-uniform. The majority of regions display minimal demand, while a small number of areas indicate travel requirements. The model effectively captures this feature by accurately identifying regions with high demand and providing precise predictions for these areas.

To provide a more in-depth analysis of prediction performance, we deliberately selected two OD pairs characterized by the highest average demand and the lowest variance. The prediction results for their OD sequence values were visually represented, as shown in Figure 10. The graph clearly demonstrates that the model closely approximates the actual data, effectively capturing the distinctive OD demand patterns across various time intervals. The prediction performance is notably accurate and robust.

Ablation Experiments

To further assess the logical basis and benefits of the diverse components integrated into the proposed RF-STED model, a series of ablation experiments were carried out using the RF-STED framework. These experiments involved manipulating the model's structure, reducing the number of modules in the RF-STED model, adjusting model parameters, and other similar strategies to establish comparative ablation models. As presented in the following list, the ablation models were then utilized to predict the dataset, and various evaluation metrics were computed for each ablation model. The resulting predictive results are summarized in Table 2.

By analyzing the prediction results of the RF-STED-No ED No Encoder Decoder module model, it is evident that a slight decline in prediction accuracy occurred. This suggests that the encoding layer had a positive impact on the primary model's prediction. Moreover, the presence of the encoding layer enabled the primary model to effectively fuse spatial and temporal features, contributing to enhanced prediction accuracy.

By analyzing the prediction results of the RF-STED-No RF No Residual Feature Extractor Module model, it is apparent that there was a notable decrease in prediction accuracy compared with both the primary model and the RF-STED-No ED model. This underscores the positive role played by the residual feature extractor in the primary model. The residual convolutional network applies multiple convolution operations to the vector with spatial and temporal features obtained from the encoding layer. This aids the model in extracting relationships between temporal and spatial features, thereby elevating prediction accuracy.

By analyzing the prediction results of the RF-STED-No S No spatial feature extraction module model, it can be seen that the prediction accuracy of the model has significantly decreased, indicating the necessity of the spatial feature extraction module. Also, it proves that the spatial relationship between the four OD pairs proposed in this article is reasonable and correct. At the same time, analyzing the prediction results of the RF-STED-No T No time feature extraction module model, it can be seen that the prediction accuracy of the model has significantly decreased, indicating the necessity and importance of the

Table 1. Prediction Result of Each Model

	OD category—two			OD category—three			OD category—four		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
ARIMA	8.01	6.32	0.78	12.10	8.54	0.80	3.75	2.29	0.69
2D CNN	7.31	5.52	0.71	11.76	8.59	0.85	4.08	2.72	0.90
3D CNN	9.17	7.18	1.00	12.82	9.47	1.00	4.28	2.86	1.00
LSTM	8.43	6.45	0.81	12.11	8.81	0.84	3.70	2.41	0.66
GRU	8.42	6.43	0.79	12.10	8.80	0.82	4.28	2.86	0.99
Conv-LSTM	3.89	2.89	0.59	4.80	3.54	0.53	2.48	1.64	0.53
GCN	7.03	5.14	0.69	11.04	8.33	0.80	3.91	2.66	0.89
S-ResNet	9.10	7.12	0.99	12.64	9.33	0.97	4.29	2.90	1.01
ST-ResNet	9.23	7.23	1.01	12.80	9.45	0.98	4.19	2.81	0.94
T-GCN	6.53	3.99	0.75	10.05	6.76	0.77	3.64	2.04	0.60
RF-STED	**2.88**	**2.22**	**0.35**	**3.72**	**2.67**	**0.36**	**1.74**	**1.12**	**0.31**

Note: ARIMA = Autoregressive Integrated Moving Average; Conv-LSTM = Convolutional Long Short-Term Memory; GCN = Graph Convolutional Network; GRU = Gated Recurrent Unit; LSTM = Long Short-Term Memory; S-ResNet = Spatial Residual Network; ST-ResNet = Spatiotemporal Residual Network; T-GCN = Temporal Graph Convolutional Network; 2D CNN = Two-Dimensional Convolutional Neural Network; 3D CNN = Three-Dimensional Convolutional Neural Network; RF-STED = Spatiotemporal Encoder-Decoder Network with a Residual Feature Extractor.
Bold represents the performance of the model proposed in my paper.

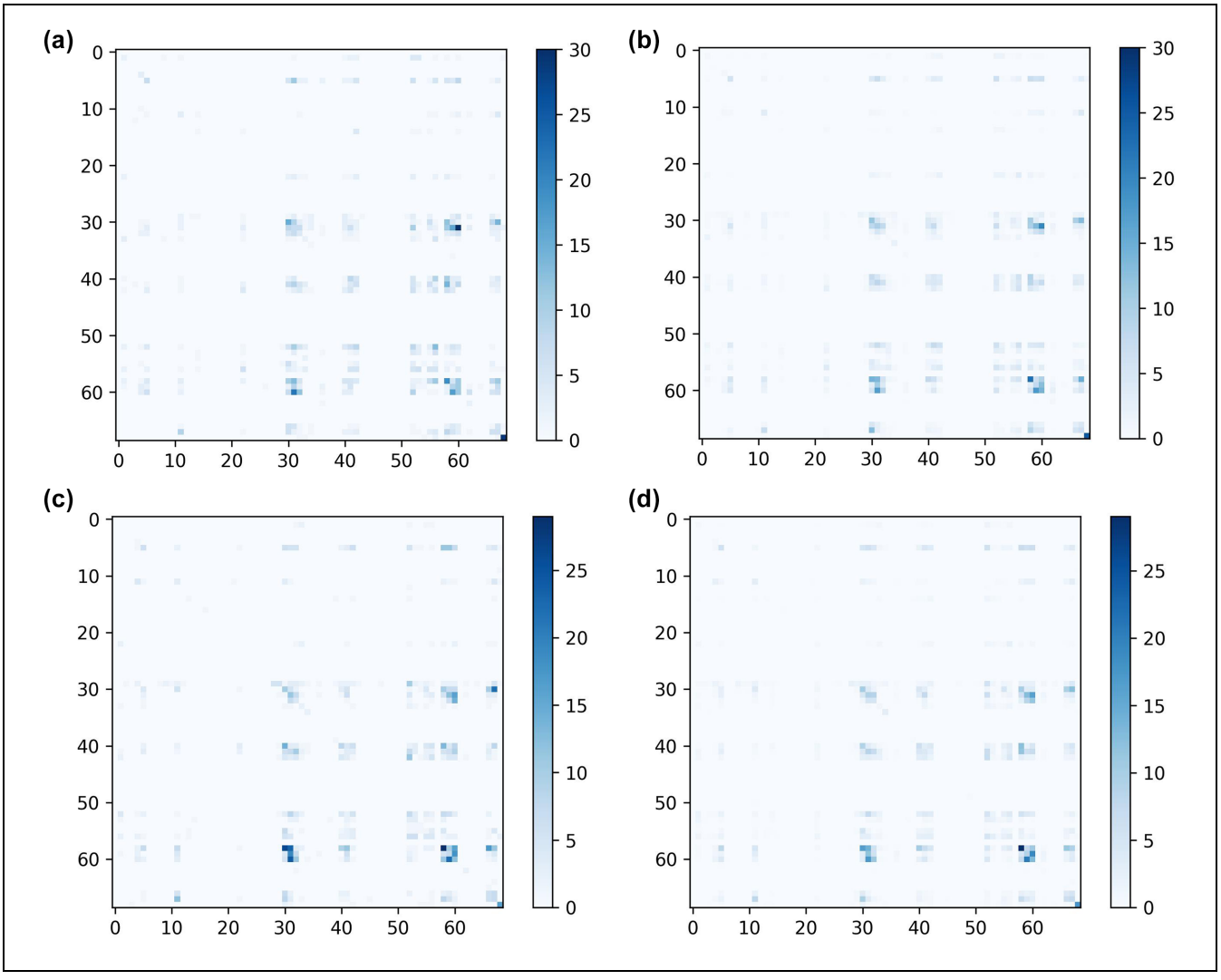

Figure 9. The prediction result of origin-destination distribution: (*a*) true (1.4–18:00), (*b*) prediction (1.4–18:00), (*c*) true (1.8–11:00), and (*d*) prediction (1.8–11:00).
The x and y axes represent the starting and ending region IDs, with values representing the od quantity.

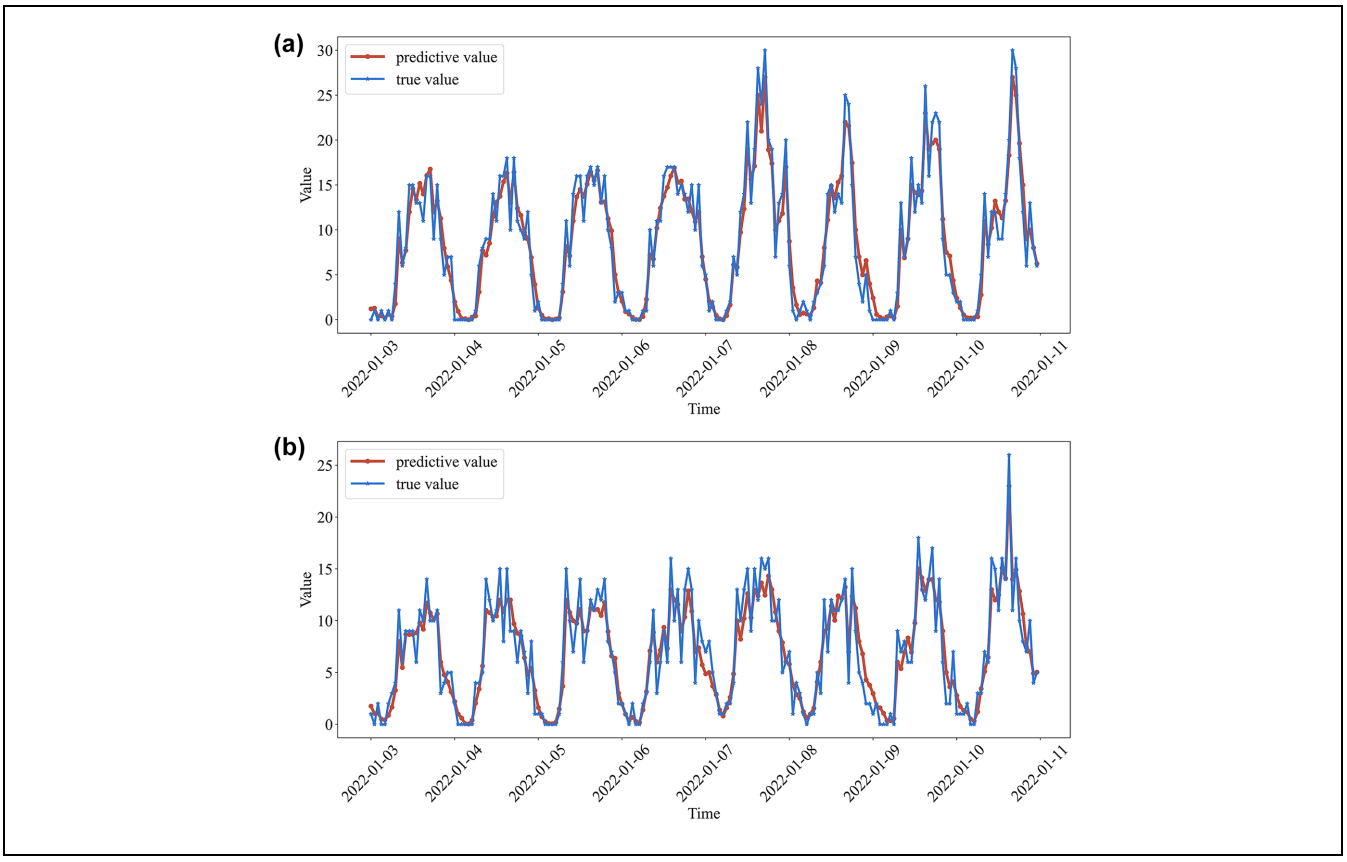

Figure 10. The prediction result of different origin-destination (OD) pairs over time: (*a*) OD pair *a* and (*b*) OD pair *b*.

Table 2. Prediction Result of Ablation Experiments

	OD category—two			OD category—three			OD category—four		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
RF-STED	**2.88**	**2.22**	**0.35**	**3.72**	**2.67**	**0.36**	**1.74**	**1.12**	**0.31**
RF-STED-No ED	2.97	2.29	0.45	3.74	2.71	0.43	1.78	1.15	0.33
RF-STED-No RF	3.04	2.34	0.47	3.86	2.75	0.45	1.79	1.16	0.33
RF-STED-No S	3.73	2.85	0.52	4.78	3.40	0.51	2.28	1.44	0.35
RF-STED-No T	7.16	5.53	0.57	10.26	7.58	0.56	2.34	1.54	0.39

Note: MAE = Mean Absolute Error; MAPE = Mean Absolute Percentage Error; RMSE = Root Mean Square Error.
Bold represents the performance of the model proposed in my paper

time feature extraction module. By comparing with the RF-STED-No S model, it can be seen that the temporal characteristics of OD pairs are more important than the spatial relationships between OD pairs, and are also an integral part of the main model.

Conclusion

In the context of the era characterized by artificial intelligence and big data, this paper establishes a deep learning framework for short-term taxi OD flow prediction, addressing the challenges of real-time and accurate prediction in the context of taxi OD flows. This framework provides a basis for dispatch decisions for ride-hailing platform personnel, thereby fostering the establishment of intelligent ride-hailing platforms. This contributes to enhancing passenger satisfaction in taxi and ride-hailing services. The main findings and contributions of this research can be summarized as follows.

Firstly, this paper introduces the concept of trend relationship graphs among OD pairs to depict spatial relationships between OD pairs. Building on this, a

classification study based on K-means clustering is conducted for OD pairs, yielding results that confirm the existence of distinctive spatial relationships among different OD pairs. The incorporation of such spatial relationships into the deep learning framework enhances prediction accuracy.

Secondly, this paper proposes a new feature extraction module—residual feature extractor—which uses convolutional check vectors of different sizes for feature extraction. The results show that the residual feature extractor performs well in the feature extraction of encoding vectors.

Thirdly, this paper proposes a deep learning framework for extracting both temporal and spatial features. Notably, the extraction of temporal features departs from the conventional LSTM network and adopts the Conv-LSTM network to better suit the format of OD data, preserving the inherent structural characteristics of OD data. The introduction of logical spatial relationships among OD pairs enriches the spatial feature extraction process. Additionally, the flexibility of the deep network framework allows for the extraction of any number of time and space features. The results demonstrate that the proposed deep learning framework significantly outperforms traditional models and individual deep learning networks in prediction accuracy.

However, there are still some shortcomings in this paper, and the main summary is as follows.

Firstly, because of limited data sources and the real-time nature of our short-term OD demand prediction, the model underwent testing solely on a subset of taxi datasets in Manhattan during January. Consequently, the model's performance has not been validated across different seasons. Additionally, owing to constraints in computing resources, we restricted the input data to 3 days of historical information. However, it is noteworthy that our model is not confined to this limitation. Given ample computing resources, it can be trained and tested on more extensive datasets covering a more extended period, such as 10 years. As part of future endeavors, the model's predictive capabilities could be enhanced by incorporating additional external features, including weather, temperature, and emergency situations. This expanded set of characteristics has the potential to contribute to more robust predictions.

Secondly, this study is based on normalized taxi OD data under normal circumstances. The research does not delve into short-term taxi OD flows related to major events or unexpected incidents. Such predictions exhibit greater stochasticity and necessitate more complex model constructions. Future research can focus on exploring these types of passenger flow.

Thirdly, while the constructed deep learning framework demonstrates effective predictive performance for existing taxi data, because of experimental constraints, the model has not been applied in a continuous learning setting or real-world practice. Future endeavors could involve the practical application of the model in real-time training and prediction scenarios.

Acknowledgment

We would like to thank the Taxi and Limousine Commission website for providing relevant taxi order data and regional division maps.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: X. Zhong, J. Zhang; data collection: J. Zhang, Q. Hua; analysis and interpretation of results: X. Zhong, J. Zhang, L. Yang, Z. Gao; draft manuscript preparation: X. Zhong. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (Nos. 72201029, 72288101, 72322022).

ORCID iDs

Jinlei Zhang https://orcid.org/0009-0002-8864-0725
Qiang Hua https://orcid.org/0000-0001-9017-0043
Lixing Yang https://orcid.org/0000-0003-1628-5015

References

1. Ke, J., X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye. Predicting Origin-Destination Ride-Sourcing Demand with a Spatio-Temporal Encoder-Decoder Residual Multigraph Convolutional Network. *arXiv Preprint arXiv:1910.09103*, 2021.
2. Vazifeh, M. M., P. Santi, G. Resta, S. H. Strogatz, and C. Ratti. Addressing the Minimum Fleet Problem in On-Demand Urban Mobility. *Nature*, Vol. 557, No. 7706, 2018, pp. 534–538.
3. Liu, Y., and Y. Li. Pricing Scheme Design of Ridesharing Program in Morning Commute Problem. *Transportation Research Part C: Emerging Technologies*, Vol. 79, 2017, pp. 156–177.
4. Zhang, D., F. Xiao, M. Shen, and S. Zhong. DNEAT: A Novel Dynamic Node-Edge Attention Network for Origin-Destination Demand Prediction. *Transportation*

Research Part C: Emerging Technologies, Vol. 122, 2021, p. 102851.
5. Deng, Z., and M. Ji. Spatiotemporal Structure of Taxi Services in Shanghai: Using Exploratory Spatial Data Analysis. *Proc., 19th International Conference on Geoinformatics*, Shanghai, China, IEEE, New York, 2011, pp. 1–5.
6. Moreira-Matias, L., J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. Predicting Taxi-Passenger Demand Using Streaming Data. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 3, 2013, pp. 1393–1402.
7. Liu, L. A Novel Passenger Flow Prediction Model Using Deep Learning Methods. *Transportation Research Part C: Emerging Technologies*, Vol. 84, 2017, pp. 74–91.
8. Zhang, J., and F. Chen. Short-Term Origin-Destination Forecasting in Urban Rail Transit Based on Attraction Degree. *IEEE Access*, Vol. 7, 2019, pp. 133452–133462.
9. Ma, D. Input Data Selection for Daily Traffic Flow Forecasting Through Contextual Mining and Intra-Day Pattern Recognition. *Expert Systems with Applications*, Vol. 176, 2021, p. 114902.
10. Geng, X., Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 3656–3663.
11. Goyal, P., S. R. Chhetri, and A. Canedo. Dyngraph2vec: Capturing Network Dynamics Using Dynamic Graph Representation Learning. *Knowledge-Based Systems*, Vol. 187, 2020, p. 104816.
12. Yang, Y., J. Zhang, L. Yang, Y. Yang, X. Li, and Z. Gao. Short-Term Passenger Flow Prediction for Multi-Traffic Modes: A Transformer and Residual Network Based Multi-Task Learning Method. *Information Sciences*, Vol. 642, 2023, p. 119144.
13. Dong, Y., S. Wang, L. Li, and Z. Zhang. An Empirical Study on Travel Patterns of Internet-Based Ride-Sharing. *Transportation Research Part C: Emerging Technologies*, Vol. 86, 2018, pp. 1–22.
14. Mayer, H. M. Final Report, Chicago Area Transportation Study. *Economic Geography*, Vol. 37, No. 4, 1961, pp. 373–374.
15. Shaygan, M., C. Meese, W. Li, X. Zhao, and M. Nejad. Traffic Prediction Using Artificial Intelligence: Review of Recent Advances and Emerging Opportunities. *Transportation Research Part C: Emerging Technologies*, Vol. 145, 2022, p. 103921.
16. Zhang, J. *Research on Short Term Passenger Flow Prediction Method for Urban Rail Transit Network*. Beijing Jiaotong University, China, 2021.
17. Yao, X. M., P. Zhao, and D. D. Yu. Dynamic Origin-Destination Matrix Estimation for Urban Rail Transit Based on Averaging Strategy. *Journal of Jilin University (Engineering and Technology Edition)*, Vol. 46, No. 1, 2016, pp. 92–99.
18. Yao, X. M., P. Zhao, and D. D. Yu. OD Estimation Model for Short-Term Passenger Flow in Urban Rail Transit Network. *Transportation Systems Engineering and Information*, 2015, Vol. 11, pp. 149–155.
19. Xia, L., Y. Jie, C. Lei, and C. Ming-Rui. Prediction for Air Route Passenger Flow Based on a Grey Prediction Model. *Proc., 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CYBERC)*, Chengdu, China, IEEE, New York, 2016, pp. 185–190.
20. Run, L., L. X. Min, and Z. X. Lu. Research and Comparison of ARIMA and Grey Prediction Models for Subway Traffic Forecasting. *Proc., 2020 International Conference on Intelligent Computing and Automation Systems (ICICAS)*, Chongqing, China, IEEE, New York, 2020, pp. 63–67.
21. Ma, X., Z. Tao, Y. Wang, H. Yu, and Y. Wang. Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data. *Transportation Research Part C: Emerging Technologies*, Vol. 54, 2015, pp. 187–197.
22. Jiang, F., F. Jia, and J. Feng. Online Dynamic Estimation of Passenger Flow OD in Urban Rail Network Based on AFC Data. *Transportation Systems Engineering and Information*, Vol. 18, No. 5, 2018, pp. 129–135.
23. Bai, S., J. Z. Kolter, and V. Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv Preprint arXiv:1803.01271*, 2018.
24. Xiao, F., D. Zhang, G. Kou, and L. Li. Learning Spatiotemporal Features of Ride-Sourcing Services with Fusion Convolutional Network. *arXiv Preprint arXiv:1904.06823*, 2019.
25. Krizhevsky, A., I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Proc., 26th International Conference on Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, Curran Associates, Inc., Red Hook, NY, USA, 2012, pp. 1097–1105.
26. Liu, L., Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin. Contextualized Spatial-Temporal Network for Taxi Origin-Destination Demand Prediction. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 10, 2019, pp. 3875–3887.
27. Chai, D., L. Wang, and Q. Yang. Bike Flow Prediction with Multi-Graph Convolutional Networks. *Proc., 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle, WA, Association for Computing Machinery, New York, 2018, pp. 397–400.
28. Sun, J., J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng. Predicting Citywide Crowd Flows in Irregular Regions Using Multi-View Graph Convolutional Networks. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 34, No. 5, 2022, pp. 2348–2359.
29. Shi, X., Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Proc., 29th International Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, MIT Press, Cambridge, MA, 2015, pp. 802–810.
30. Jiang, J., F. Lin, J. Fan, H. Lv, and J. Wu. A Destination Prediction Network Based on Spatiotemporal Data for Bike-Sharing. *Complexity*, Vol. 2019, 2019, Article ID 14.

31. Yu, B., H. Yin, and Z. Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proc., 27th International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 3634–3640.
32. Wu, Z., S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, No. 1, 2021, pp. 4–24.
33. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, A. Kaiser, and I. Polosukhin. Attention Is All You Need. *Proc., 31st Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, Curran Associates, Inc., Red Hook, NY, 2017, pp. 5998–6008.
34. Wang, F., M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual Attention Network for Image Classification. *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Xplore, 2017, pp. 3156–3164.
35. Yang, Z., D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical Attention Networks for Document Classification. *Proc., 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, San Diego, CA, Association for Computational Linguistics, 2016, pp. 1480–1489.
36. Chen, L., H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. Chua. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Xplore, 2017, pp. 5659–5667.
37. Sankar, A., Y. Wu, L. Gou, W. Zhang, and H. Yang. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. *Proc., 13th ACM International Conference on Web Search and Data Mining*, Houston, TX, Association for Computing Machinery, New York, 2020, pp. 519–527.
38. Zhang, J., H. Che, F. Chen, W. Ma, and Z. He. Short-Term Origin-Destination Demand Prediction in Urban Rail Transit Systems: A Channel-Wise Attentive Split-Convolutional Neural Network Method. *Transportation Research Part C: Emerging Technologies*, Vol. 124, 2021, p. 102928.
39. Zhang, S., J. Zhang, L. Yang, J. Yin, and Z. Gao. Spatiotemporal Attention Fusion Network for Short-Term Passenger Flow Prediction on New Year's Day Holiday in Urban Rail Transit System. *IEEE Intelligent Transportation Systems Magazine*, Vol. 15, No. 5, 2023, pp. 59–77.
40. Zhang, J., Y. Zheng, and D. Qi. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31, No. 1, 2017, pp. 1655–1661.
41. MacQueen, J. B. Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, 1967, pp. 281–297.
42. Kipf, T. N., and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv Preprint arXiv:1609.02907*, 2016.
43. Levie, R., F. Monti, X. Bresson, and M. M. Bronstein. CayleyNets: Graph Convolutional Neural Networks with Complex Rational Spectral Filters. *IEEE Transactions on Signal Processing*, Vol. 67, No. 1, 2017, pp. 97–109.
44. Li, R., S. Wang, F. Zhu, and J. Huang. Adaptive Graph Convolutional Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, No. 1, 2018, pp. 3546–3553.
45. Atwood, J., and D. Towsley. Diffusion-Convolutional Neural Networks. *Proc., 30th Conference on Neural Information Processing Systems (NIPS)*, Barcelona, Spain, Curran Associates, Inc., Red Hook, NY, USA, 2016, pp. 1993–2001.
46. Niepert, M., M. Ahmed, and K. Kutzkov. Learning Convolutional Neural Networks for Graphs. *Proceedings of the International Conference on Machine Learning, PMLR*, Vol. 48, 2016, pp. 2014–2023.
47. Hochreiter, S., and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780.
48. He, K., X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385, 2015. https://arxiv.org/abs/1512.03385
49. LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2278–2324.
50. Tran, D., L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. *Proc., 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, IEEE, New York, 2015, pp. 4489–4497.
51. Cho, K., and B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. *Proc., 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Association for Computational Linguistics, 2014, pp. 1724–1734.
52. Yu, B., and Y. Lee. Forecasting Road Traffic Speeds by Considering Area-Wide Spatiotemporal Dependencies Based on a Graph Convolutional Neural Network. *Transportation Research Part C: Emerging Technologies*, Vol. 114, 2020, pp. 189–204.
53. Silberman, N., L. Shapira, R. Gal, and P. Kohli. Temporal Residual Networks for Dynamic Scene Recognition. *Proc., 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, IEEE, New York, 2017, pp. 5513–5521.