

Article

Prediction of Inbound and Outbound Passenger Flow in Urban Rail Transit Based on Spatio-Temporal Attention Residual Network

Jun Yang^{1,2}, Xueru Dong^{1,*}, Huifan Yang¹, Xiao Han¹, Yan Wang¹ and Jiayue Chen¹

¹ Big Data and Internet of Things Research Center, China University of Mining and Technology-Beijing, Beijing 100083, China

² Key Laboratory of Intelligent Mining and Robotics, Ministry of Emergency Management, Beijing 100083, China

* Correspondence: xueru_dong@163.com

Abstract: Passenger flow prediction is a critical approach to ensure the effective functioning of urban rail transit. However, there are few studies that combine multiple influencing factors for short-term passenger flow prediction. It is also a challenge to accurately predict passenger flow at all stations in the line at the same time. To overcome the above limitations, a deep learning-based method named ST-RANet is proposed, which consists of three spatio-temporal modules and one external module. The model is capable of predicting inbound and outbound passenger flow for all stations within the network simultaneously. We model the spatio-temporal data in terms of three temporal characteristics, including closeness, period, and trend. For each characteristic, we construct a spatio-temporal module that innovatively integrates the attention mechanisms into the middle of residual units and convolutional neural networks (CNNs) to extract and learn spatio-temporal features. Subsequently, the results of the three modules are integrated using a parameter matrix method, which allows for dynamic aggregation based on data. The integration results are further combined with external factors, such as holidays and meteorological information, to obtain passenger flow prediction values for each station. The proposed model is validated using real data from Beijing Subway, and optimized parameters are applied for 30-min granularity passenger flow predictions. Comparing the performance against 5 baseline models and verifying with data from multiple lines, the results indicate that the proposed ST-RANet model shows the best results. It is demonstrated that the method proposed in this paper has high prediction accuracy and good applicability.

Keywords: passenger flow prediction; attention mechanism; residual network; deep learning; urban rail transit



Citation: Yang, J.; Dong, X.; Yang, H.; Han, X.; Wang, Y.; Chen, J. Prediction of Inbound and Outbound Passenger Flow in Urban Rail Transit Based on Spatio-Temporal Attention Residual Network. *Appl. Sci.* **2023**, *13*, 10266.

<https://doi.org/10.3390/app131810266>

Academic Editors: Fei Xiong, Shirui Pan, Hongshu Chen and Xuzhen Zhu

Received: 19 August 2023

Revised: 9 September 2023

Accepted: 12 September 2023

Published: 13 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the past few years, steady growth of urbanization and rapid economic development have led to the sustained expansion of urban scale, resulting in an increasingly growing demand for travel. Urban rail transit, as a new green, and efficient type of public transportation, has become an important component of transportation due to its continuous growth and improvement in large and medium-sized cities. Under such conditions, passenger flow prediction has been one of the pivotal technologies for ensuring the uninterrupted operations and optimal planning of rail transit systems [1]. Highly accurate short-term traffic forecasting can aid operators in optimizing resource utilization, including proactively managing passenger flow [2], and modifying train schedules well in advance [3,4]. Moreover, it can help anticipate station congestion, effectively ensuring the safety of passengers and safeguarding their assets [5]. In short, the short-term passenger flow prediction [6] for urban rail transit is of great importance in the fields of daily functioning, refined management planning, and public safety.

The development process of rail transit passenger flow prediction can be traced back to the 1970s. The initial prediction methods are mainly based on statistical models to predict future passenger flow trends by analyzing historical data [7]. The Historical Average (HA) model [8] forecasts future values based on the average of past observations. The autoregressive integrated moving average model (ARIMA) [9], as described, combines three steps: the autoregressive, integrated, and moving average, which can flexibly adapt to different time series data and obtain the optimal prediction results by adjusting parameters. Seasonal autoregressive integrated moving average (SARIMA) [10] is an expansion of the ARIMA model which captures seasonal effects by adding additional seasonal differences and seasonal autoregressive terms. However, both require complex parameter selection for the model, and they perform poorly when there are multiple seasonal patterns in the time series. Although statistical methods are simple and feasible, their accuracy is low and cannot meet practical needs because they cannot consider the complexity of the influencing factors and the uncertainty of the data.

The development of computer technology has paved the way for the use of innovative techniques like big data and machine learning in predicting passenger flow for rail transit systems [11]. This includes some models like back-propagation neural networks (BPNNs) [12], support vector machines (SVM) [13], and random forest-learning [14], among others. Among them, deep learning, as a subset of machine learning, can automatically discover complex relationships and patterns within data and can be trained on large-scale data. Liao et al. [15] predicted the parameters of a mathematical model of infectious diseases by fusing deep learning models with other temporal prediction methods. Zhu et al. [16] fused several deep learning models to propose a new knee angle prediction model. Polson et al. [17] developed a deep learning model to predict traffic flow. Numerous research results [18,19] demonstrate that deep learning methods provide a good idea to improve prediction accuracy.

Recurrent neural network (RNN) is a type of deep learning model that is well-suited for modeling sequential data and can effectively handle passenger flow data with temporal dependencies [20]. Long short-term memory (LSTM) is a kind of RNN capable of capturing both short-term and long-term dependencies and relationships between variables, making it highly effective in passenger flow prediction tasks [21,22]. In 2014, Cho et al. [23] proposed a lightweight variant of LSTM called gated recurrent unit (GRU), which uses fewer training parameters than LSTM and can achieve comparable or even better results. Concurrently, this model also reduces the training time and computational costs. Subsequently, many researchers have investigated prediction models that combine multiple methods. This is because only RNN models cannot capture the complex data characteristics well, resulting in poor prediction accuracy. Yang et al. [24] introduced a novel Wave-LSTM approach that incorporates wavelets and LSTM for predicting inbound passenger flows in urban rail transit. Dai et al. [25] integrated spatio-temporal analysis with GRU for passenger flow prediction, and proved its superior accuracy and stability compared to the standalone GRU model. Zhang et al. [26] used K-means clustering to catch the characteristics of passenger flow and subsequently fed them into an LSTM model to achieve short-term passenger flow prediction. However, these studies have some drawbacks in that they cannot predict multiple sites within the network simultaneously due to the inability to capture the spatial correlation.

Many researchers use deep neural networks (DNN) in rail transit passenger flow prediction tasks, which are capable of acquiring spatio-temporal information and topological information. Wu et al. [27] proposed a DNN-based traffic flow prediction model (DNN-BTF) and demonstrated its ability to predict traffic flow using big data. Combining CNN [28] can also automatically extract some important spatial features. Zhang et al. [29] introduced a CNN-based forecasting model named DeepST for urban pedestrian flow prediction. They separated the temporal characteristics of pedestrian flow into three modules: closeness, period, and trend, combined with spatial features input to the CNN, and considered the influence of holidays and meteorological information. This method com-

prehensively captures the elemental features affecting pedestrian flow and achieves good prediction performance. In 2015 He et al. [30] proposed Deep Residual Learning, which addresses the gradient disappearance and explosion problem by introducing “residual connections”. It has been effectively utilized in diverse datasets and applications, particularly in road traffic scenarios. Based on this, Zhang et al. [31] introduced an upgraded version of DeepST, named ST-ResNet, which leveraged residual network architecture to enhance its spatio-temporal modeling capabilities. After validation on bicycle and taxi datasets, the results proved to be better than DeepST. However, this excellent method has not been applied to rail passenger flow prediction scenarios yet, which brings inspiration for the generation of this article.

As deep learning techniques evolve, attention mechanisms [32] have become a hot topic for researchers. The basic principle of the attention mechanism is to make the model pay more attention to important information by assigning different weights to different parts or features of the input. Through the attention mechanism, the model can automatically learn which parts or features are more important for the current task, thus improving the performance and effectiveness of the model. In recent years, it has made significant progress in various fields. Bello et al. [33] considered the use of the attention mechanism for discriminative vision tasks, and experiments demonstrated that the attention mechanism achieved improvements on many models, in image classification on ImageNet and object detection on COCO. In the problem of time series data prediction, the addition of the attention mechanism can help the model to better handle time series data. Qin et al. [34] used a two-stage attention-based recurrent neural network (DA-RNN) to capture long-term temporal correlations, outperforming state-of-the-art methods in time series prediction. Yang et al. [35] introduced a localized spatio-temporal neural network (STNN), which employs novel spatio-temporal convolutional and attentional mechanisms to learn universal spatio-temporal correlations and improve traffic flow prediction accuracy.

The above describes the evolution of passenger flow forecasting methods from traditional methods to artificial intelligence methods. It reflects the fact that researchers are constantly seeking methods to mine the multiple aspects of information that influence prediction. In the context of urban rail transit, there is a relative lack of studies that combine multiple temporal and spatial influences for short-term passenger flow forecasting. It is particularly challenging to accurately predict the passenger flow at each station on a given line. To address this research gap, this paper aims to propose a deep learning-based method for predicting short-term inbound and outbound passenger flows in urban rail transit, called deep spatio-temporal residual attention network (ST-RANet), to improve the accuracy and precision of the predictions. ST-RANet combines the attention mechanisms with residual units and convolutional networks, to extract both multi-temporal and spatial features of passenger flow. In addition to considering the spatio-temporal correlation between stations, some external factors that affect passenger travel are also included in ST-RANet, to determine passenger flow trends with multiple considerations. We have evaluated the performance of our proposed ST-RANet model against multiple advanced models that are widely utilized for predicting passenger flow in railway transit systems. The results of our experiments indicate that our ST-RANet model outperformed all other baseline models in terms of prediction accuracy.

The main contributions of this paper can be summarized as follows:

- (1) Based on the characteristics of passenger flow at different timescales, the spatio-temporal modules are divided into three categories: closeness, period, and trend. What's more, the model innovatively introduces a convolutional block attention module (CBAM) to allocate different weights for three spatio-temporal modules, enabling deeper mining of spatio-temporal data features.
- (2) The introduced ST-RANet model is able to predict rail transit passenger flow at all stations of the transportation network simultaneously, not only considering the spatio-temporal characteristics of rail transit passenger flow but also combining holiday and meteorological data to achieve comprehensive and accurate real-time predictions.

- (3) By utilizing real inbound and outbound passenger flow data from Beijing’s metro system to evaluate ST-RANet, this study discovers that various evaluation metrics indicate an enhancement in prediction accuracy by ST-RANet, as compared to the baseline model.

The rest of this paper is organized as follows: Section 2 provides the problem definition and data modeling. Section 3 gives a detailed description of the structure of ST-RANet. Section 4 describes the data, parameters, and metric settings required for the experiments. Section 5 conducts the evaluation and discussion of the experimental results. Section 6 summarizes the contributions and limitations of the study and suggests directions for future work.

2. Problem Definition

In this part, we provide a specific problem description and model the core part of our research—spatio-temporal characteristics, to support the presentations of the subsequent model.

2.1. Problem Description

This study focuses on passenger flow prediction in multiple stations of urban rail transit, using the historical swipe card data of the AFC system to predict the future inbound and outbound traffic simultaneously. That is, for a given target time slice t , the spatio-temporal flow data of the past n time slices are taken as input for the prediction model, and the predicted value for the time slice t is the output. We built a spatial matrix within each time interval using the location information of stations and then converted the inbound and outbound passenger flow into a matrix of dual channels. The real passenger flow at time slice t can be expressed as a tensor $X_t = (X_t^{(in)}, X_t^{(out)})^T \in R^{2 \times I \times J}$, where $X_t^{(in)}, X_t^{(out)}$ are the real inbound and outbound volumes respectively, $I \times J$ represents the spatial area size. $Y_t = (Y_t^{(in)}, Y_t^{(out)})^T \in R^{2 \times I \times J}$ denotes the predicted passenger flow at time slice t , $Y_t^{(in)}, Y_t^{(out)}$ are the predicted inbound and outbound traffic, respectively. In general, the urban rail transit passenger flow prediction problem analyzed in this study can be formulated by:

$$Y_t = F((X_{t-n}, \dots, X_{t-1})) \tag{1}$$

where F is the mapping function, and $(X_{t-n}, \dots, X_{t-1})$ is the historical passenger flow tensor.

2.2. Multi-Timescale Spatio-Temporal Characteristic Modeling

To fully explore the implied periodicity and trend information within passenger flow time series data, the data are divided into three distinct groups according to the timeline: near-term, mid-term, and distant-term. This division allows for the extraction of three temporal characteristics: closeness, period, and the trend of historical passenger flow data. Assuming that the daily total duration is evenly divided into T time steps, for a target predicting time slice t , the time segments $X_{closeness}^{(0)}, X_{period}^{(0)}$ and $X_{trend}^{(0)}$ are respectively extracted along the timeline with three timescales, and the number of time slices contained in each time segment is denoted by variables c, d , and w . $X_{closeness}^{(0)}, X_{period}^{(0)}$ and $X_{trend}^{(0)}$ represent the historical passenger flow tensor corresponding to each time segment, which will be used as inputs of the spatio-temporal module. The process of constructing the spatio-temporal passenger flow feature matrix is as follows.

- (1) The tensor received by the closeness module is given by $X_{closeness}^{(0)}$

$$X_{closeness}^{(0)} = (X_{t-c}, X_{t-(c-1)}, \dots, X_{t-1}) \tag{2}$$

where $X_{closeness}^{(0)} \in \mathbb{R}^{2c \times I \times J}$. The closeness module uses the 2-channel passenger flow matrices of the closest c time slices to model the feature, and the position of the nearby time segment sequence $P_{closeness}$ on the time axis is shown in Figure 1.

- (2) The tensor received by the period module can be expressed as $X_{period}^{(0)}$

$$X_{period}^{(0)} = (X_{t-dT}, X_{t-(d-1)T}, \dots, X_{t-T}) \tag{3}$$

where $X_{period}^{(0)} \in \mathbb{R}^{2d \times I \times J}$. The period module input uses the same time slices from the previous d days adjacent to the time slice t , and the position of the periodic time segment sequence P_{period} on the time axis is shown in Figure 2.

- (3) The tensor received by the trend module can be expressed as $X_{trend}^{(0)}$

$$X_{trend}^{(0)} = (X_{t-7wT}, X_{t-7(w-1)T}, \dots, X_{t-7T}) \tag{4}$$

where $X_{trend}^{(0)} \in \mathbb{R}^{2w \times I \times J}$. Using the same time slices on the same day of the week in the w weeks prior to the time slice t as the trend module input, the position of the trending time segment sequence P_{trend} on the time axis is represented in Figure 3.

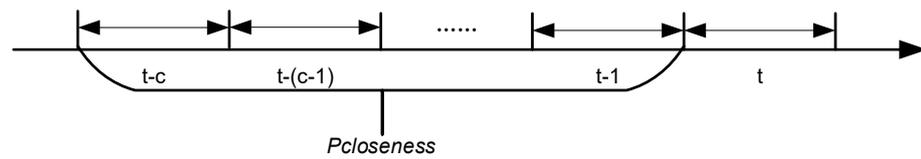


Figure 1. Time segment sequence of closeness module.

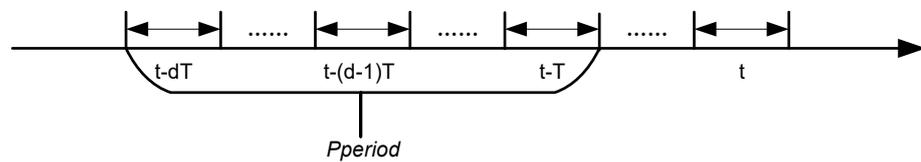


Figure 2. Time segment sequence of the period module.

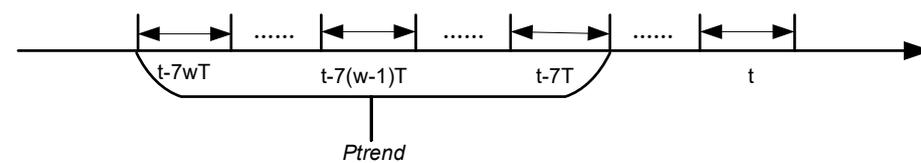


Figure 3. Time segment sequence of trend module.

3. Methodology

In this section, we propose ST-RANet and study its mechanism. Figure 4 showcases the complete design of ST-RANet, which integrates three spatio-temporal modules and an external module. The three spatio-temporal feature modules, namely closeness, period, and trend, are showcased in the upper part of the figure. The three modules use the same network to extract spatio-temporal features, consisting of two convolutional networks before and after, two CBAM attention mechanisms, and k residual learning units. The external module, as shown in the lower part of the figure, inputs holiday information and three meteorological data (including weather types, temperature, and wind power) into two fully connected layers. Subsequently, the outputs of the first three modules are fused X_{Res} according to different weights, which are further fused with the output X_{Ext} of the external module. The resulting fusion product is transformed by the hyperbolic tangent function (\tanh) to map the values onto the $[-1, 1]$ scale, and ultimately obtain the model's predicted value Y_t .

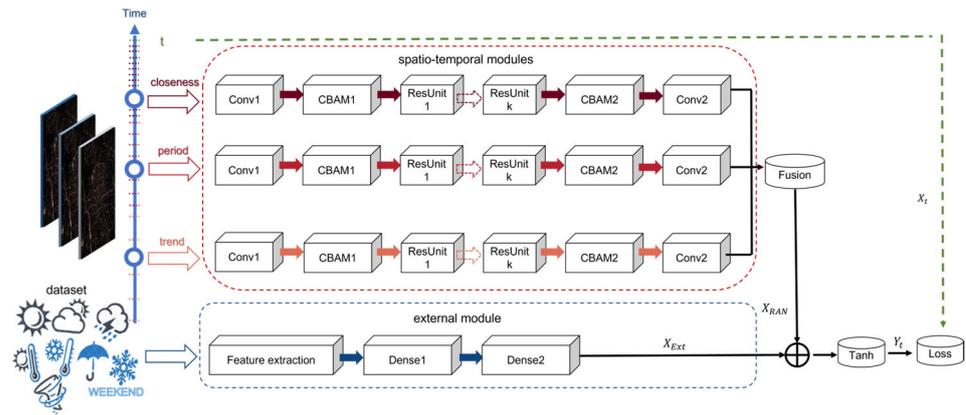


Figure 4. Architecture of ST-RANet model. Conv: convolution; ResUnit: residual unit; Dense: fully connected layer.

3.1. Spatio-Temporal Module

The network structure for the three spatio-temporal modules (closeness, period, trend) is uniform. It comprises three components namely convolution, attention mechanism, and residual unit.

3.1.1. Convolution

CNN can capture the spatial correlation of passenger flow data, and convolution operation is one of the core operations of CNN [36]. It can be understood as moving a small matrix (called convolution kernel or filter) on the input data and calculating the inner product at each position to obtain a new matrix, that contains information related to the convolution kernel in the original image.

In ST-RANet, we use only the convolution, as shown in Figure 4, using Conv1 and Conv2 at the front and back of the ST-RANet model. As depicted in Figure 5, two convolutional layers link the three feature maps. A single convolution can capture the dependency of nearby stations in space, and multiple convolutions can capture long-range dependencies even all station dependencies.

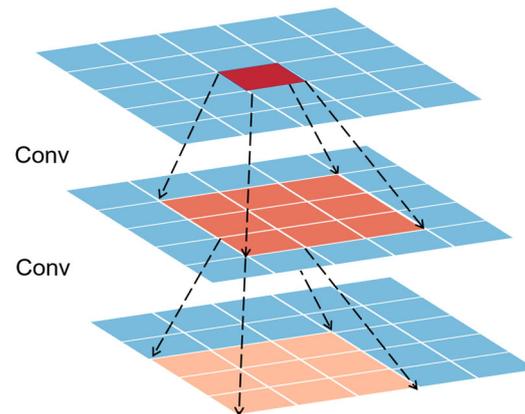


Figure 5. Convolutions.

Assuming $X^{(0)}$ is an input tensor of a spatio-temporal module, $X^{(0)} \in R^{2L \times I \times J}$, where L is the length of a time segment, which is equivalent to c , d , or w in Section 2.2. After a convolution operation (Conv1 in Figure 4), the transformation of the tensor is expressed in the following manner:

$$X^{(1)} = f(W^{(1)} * X^{(0)} + b^{(1)}) \tag{5}$$

where $X^{(1)} \in R^{C \times I \times J}$ (C is the output dimension of Conv1), f is an activation function, $*$ represents the convolution operation Conv1, $W^{(1)}$ and $b^{(1)}$ are the parameters of Conv1.

3.1.2. Convolutional Attention Mechanism Module (CBAM) [37]

CBAM is a neural network model based on attention mechanisms, which can perform attention weighting on channel and spatial dimensions respectively to extract more distinctive features in the feature maps. As a lightweight module, it uses a small number of parameters and runs fast during training, making it highly efficient. Additionally, it can be easily integrated into most of the current deep learning frameworks to enhance the feature extraction capabilities of network models.

In Figure 6, the overall architecture of CBAM is shown, with the feature map undergoing the channel attention module (CAM) and subsequently the spatial attention module (SAM) to acquire an improved feature map.

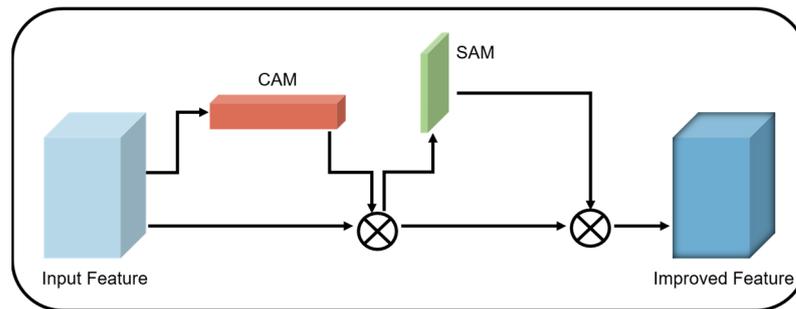


Figure 6. The architecture of CBAM.

We add CBAM after Conv1 and before Conv2 in ST-RANet (see Figure 4) to enhance the expressive power of the network. Take the first CBAM (Attention1) as an example, $X^{(1)}$ is the input. CBAM derives a channel attention map $M_c \in \mathbb{R}^{C \times 1 \times 1}$ and a spatial attention map $M_s \in \mathbb{R}^{1 \times I \times J}$ in turn. This process can be expressed by the equation,

$$\begin{aligned} Z &= M_c(X^{(1)}) \otimes X^{(1)} \\ X^{(2)} &= M_s(Z) \otimes Z \end{aligned} \tag{6}$$

where the symbol \otimes represents element-wise multiplication, and Z is an intermediate product.

The architecture of CAM is shown in Figure 7, the process of which involves obtaining the input feature map and subjecting it to global max pooling (MaxPool) as well as global average pooling (AvgPool) on its width and height dimensions. The resulting pooled feature maps can then be fed through a shared multi-layer perceptron (MLP) network. The MLP's outputs are combined by element-wise addition, and the resulting features are further processed with a sigmoid activation function to produce the final outcome map of CAM M_c . This calculation process of CAM is mathematically represented by the following equation,

$$\begin{aligned} M_c(X^{(1)}) &= \sigma(MLP(GAP(X^{(1)})) + MLP(GMP(X^{(1)}))) \\ &= \sigma(W_1(W_0(Z_{avg}^c)) + W_1(W_0(Z_{max}^c))) \end{aligned} \tag{7}$$

where σ means the sigmoid function, $W_0 \in \mathbb{R}^{C/r \times C}$ and $W_1 \in \mathbb{R}^{C \times C/r}$, in which r is the reduction ratio.

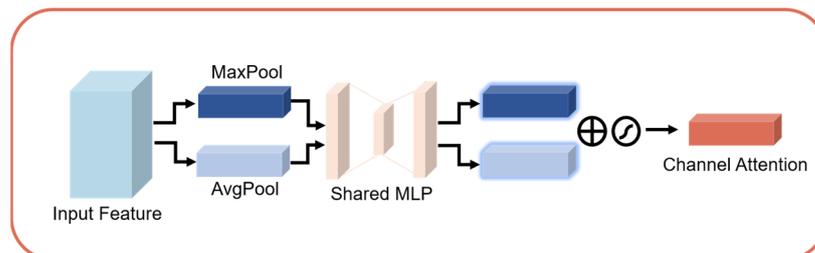


Figure 7. Architecture of CAM.

The SAM, depicted in Figure 8, starts by applying MaxPool and AvgPool operations on the input feature map along the channel axis. Subsequently, the resulting pooled feature maps are concatenated according to channels. The concatenated features are then subjected to a convolution operation which reduces the dimensionality of each spatial location to a single channel. Finally, the resulting features are passed through a sigmoid activation function to generate the outcome graph of SAM M_s .

$$M_s(Z) = \sigma(f^{7 \times 7}([AvgPool(Z); MaxPool(Z)])) = \sigma(f^{7 \times 7}([Z_{avg}^s; Z_{max}^s])) \tag{8}$$

where $f^{7 \times 7}$ denotes a convolution operation and its filter size is 7×7 .

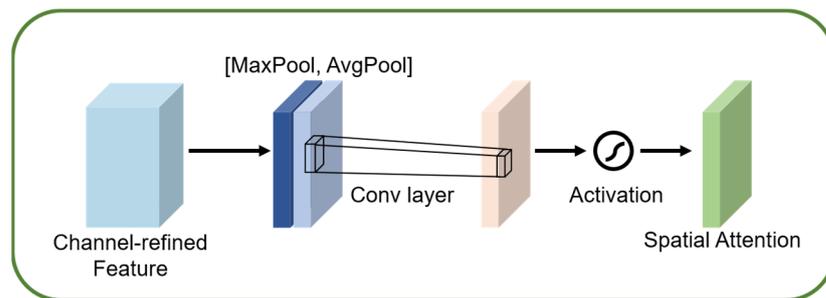


Figure 8. Architecture of SAM.

3.1.3. Residual Unit

To ensure consistency between the input and output of each layer, residual learning networks introduce residual connections, which effectively address common issues encountered in traditional neural networks including gradient disappearance or explosion. This leads to a more stable network structure that is easier to train while preserving deep feature information.

In ST-RANet (Figure 4), we apply K residual units on the output $X^{(2)}$ after Conv1 and Attention1. The formula is as follows:

$$X^{(k+2)} = X^{(k+1)} + \varphi(X^{(k+1)}; \theta^{(k)}), k = 1, \dots, K \tag{9}$$

where the input and output of the k^{th} residual unit are denoted by $X^{(k+1)}$ and $X^{(k+2)}$, respectively, the residual unit function φ is composed of a combination of rectified linear unit (ReLU) function [38] and Conv (as illustrated in Figure 9 below), and $\theta^{(k)}$ represents all the parameters of the k^{th} residual unit.

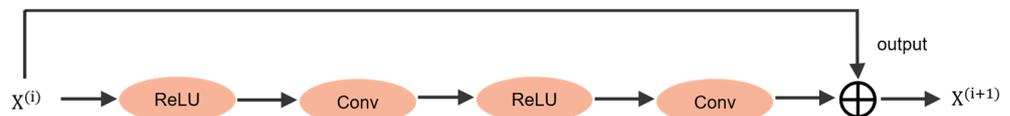


Figure 9. Residual unit.

3.1.4. Fusion of Spatio-Temporal Modules

On top of the K residual units, we add a second CBAM module (i.e., CBAM2 shown in Figure 4) and a convolution layer (i.e., Conv2 in Figure 4), and the final output is $X^{(k+4)}$. Following the same steps as above, we can construct three spatio-temporal modules of closeness, period, and trend in Figure 4. After processing through the model, the outputs of the three modules are $X_{closeness}^{(k+4)}$, $X_{period}^{(k+4)}$, and $X_{trend}^{(k+4)}$, respectively.

As the influence of closeness, period, and trend varies across different sites, we adopt a parametric-matrix-based fusion method [39] to merge the three spatio-temporal modules and obtain the output X_{RAN} :

$$X_{RAN} = W_c \circ X_{closeness}^{(k+4)} + W_p \circ X_{period}^{(k+4)} + W_q \circ X_{trend}^{(k+4)} \quad (10)$$

where W_c , W_p , and W_q are weight parameters representing the degree of influence of three modules, respectively, and \circ indicates the Hadamard product.

3.2. External Module and Fusion

To enhance the robustness and applicability of the model, we incorporate key factors that affect passenger travel in rail transit, such as holidays and meteorology. The weekday and holiday information of the time slice t can be obtained directly. As the meteorological information of the predicted time is unknown in actual situations, we use the meteorological information of the time period $t - 1$ to approximate it. Use E_t to represent the external factors of predicting time slice t , which is then inputted into a neural network comprising of two fully connected layers (see Figure 4). The primary layer is equivalent to an embedding layer for each factor, and the second layer maps the low-dimensional feature matrix to a high-dimensional matrix of the same dimension as X_t . Additionally, add a ReLU activation function after each layer and obtain the results X_{Ext} from the external module.

Finally, the results of the external module and the spatio-temporal module are merged to obtain the passenger flow prediction value Y_t for the time slice t :

$$Y_t = \text{Tanh}(X_{Res} + X_{Ext}) \quad (11)$$

where $\text{Tanh}()$ ensures that resultant values are constrained between -1 and 1 , which serves to enhance the convergence speed.

4. Experimental Settings

4.1. Datasets

The dataset utilized in this research, named SubwayBJ, consists of real data from the Beijing subway AFC system along with corresponding information on Beijing holidays and meteorological conditions. The proposed model is then trained and tested using the produced dataset.

The particulars of the SubwayBJ dataset are illustrated in Table 1. The dataset includes inbound and outbound passenger flow data collected from Beijing Metro Line 13. We selected two time periods from 15 January 2018 to 31 December 2018 and 4 January 2019 to 31 December 2019, before the novel coronavirus pneumonia. The time granularity is set to 30 min, resulting in a total of 16 stations and 30,816 time segments. The shape of the dataset is (30,816,2,4,4), where “2” represents inbound and outbound stations, and the last two digits indicate the 4×4 station distribution network. Additionally, external data including holiday and daily meteorological information for each time segment are also added. The data has a total of 57 days of holidays, 17 weather types, a temperature span of $[-12, 38]$ and a wind speed range of $[1, 61]$ m/s.

Table 1. Dataset information.

Dataset	SubwayBJ
Date span	15 January 2018–31 December 2018 4 January 2019–31 December 2019
Time	0:00–24:00
Time granularity	30 min
Space-time grid map size	4×4
Holiday numbers	57
Weather types	17

Table 1. *Cont.*

Dataset	SubwayBJ
Temperature span/°C	[−12, 38]
Wind speed span/m/s	[1, 61]

Since the tanh function (as shown in Formula (11)) is used by the ST-RANet model to limit its output within the range of $[-1, 1]$, we apply the Min-Max normalization method to prepare the input data. By doing so, we also ensure that the data range is confined to the interval of $[-1, 1]$. The corresponding formula for this normalization process can be defined as

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \times 2 - 1 \quad (12)$$

where x represents the original passenger flow value, x_{\max} denotes the maximum value of all passenger flow data, x_{\min} represents the minimum value, and x' is the resultant normalized value.

Then each time slice is traversed, and any intervals lacking information related to closeness, period, or trend were removed, resulting in a total of 25,296 time slices. Data from the last 28 days are retained for testing purposes, while the rest of the data were employed for training. Therefore, the test set size is (1344,2,4,4) and the training set size is (23,952,2,4,4), with 10% of the data in the training set being allocated for validation.

4.2. Parameter Configuration and Evaluation Metrics

We used the Python language for program development and built models based on the deep learning frameworks TensorFlow and Keras; we used the Adam optimizer for model training. In the convolutions of Conv1 and L residual units, 64 filters of size 3×3 were used, while 2 filters of size 3×3 were used in the convolution of Conv2. All experiments in our research were conducted on a machine equipped with CPU: Intel i9-13900kf, GPU: GeForce RTX 3080, and OS: Ubuntu 22.04-LTS.

In order to achieve optimal experimental results, we manually adjusted the training parameters for the experiments. For ease of viewing, we show the finalized parameter configurations in Table 2.

Table 2. Parameter configurations.

Parameter Name	Parameter Value
length of closeness sequence (c)	3
length of period sequence (d)	1
length of trend sequence (w)	1
learning rate (lr)	0.002
the number of training sessions (epochs)	500
batch size (batch_size)	32
number of residual units (k)	3

In this study, the mean squared error (MSE) is selected as the loss value. Additionally, root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are set to comprehensively evaluate the predictive capability of the model. The formulas are as follows:

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad (13)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (14)$$

$$MAE = \frac{1}{m} \sum_{i=1}^m \left| \hat{y}_i - y_i \right| \quad (15)$$

$$MAPE = \frac{100\%}{m} \sum_{i=1}^m \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (16)$$

where \hat{y}_i is the predicted value and y_i is the true value, m represents the number of the predicted samples.

Among them, MSE squares the error, making it more sensitive to the degree of deviation in prediction errors and the outliers. RMSE takes the square root of MSE to make the dimension consistent with the true value, making it easier to interpret. MAE is more evenly sensitive to errors and can better reflect prediction errors. The smaller the values of these three metrics, the more accurate the predictive results. MAPE focuses on the percentage of prediction deviation, which has high practicality for the passenger flow prediction problem that needs to consider accuracy in this study. Its range is from 0% to positive infinity, and the closer it is to 0%, the smaller the prediction error.

5. Results and Discussions

5.1. Comparison Experiments

The prediction results of our ST-RANet model for the outbound passenger flow of a station in the SubwayBJ dataset are shown in Figure 10.

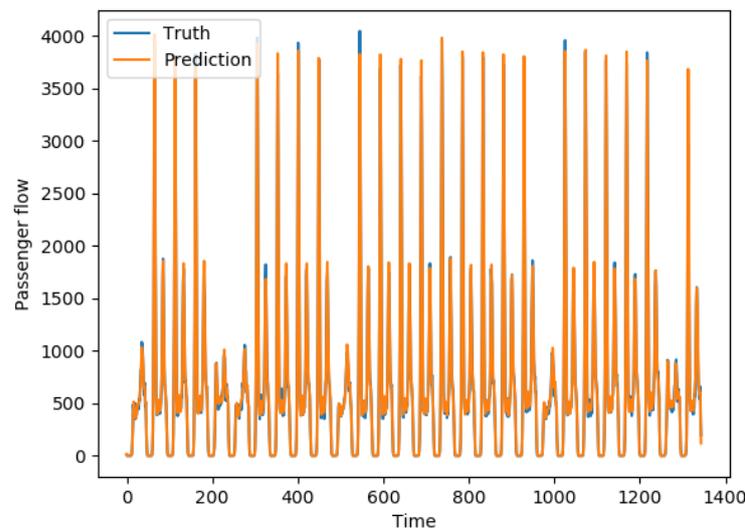


Figure 10. Prediction results of ST-RANet on SubwayBJ.

We compare the ST-RANet model with the following five baseline models using the same time periods of the dataset SubwayBJ. Their respective features and parameter settings are described below.

HA: A basic and simple forecasting method that uses the average of historical observed values as the predicted value for future values, applicable to different types of data and industries.

SARIMA: A model commonly used in time series analysis, capable of taking into account seasonality and trend. The parameters to adjust the suitable data are order = $(p,d,q) = (3,0,5)$, seasonal_order = $(P,D,Q,s) = (1,0,0,336)$, where the seasonal period is 336 that is, the number of time slices in a week.

GRU: A type of RNN that has low model complexity and high flexibility, capable of providing high prediction accuracy. The optimal training parameters are set: the batch size is 32, the number of epochs in the training stage is 1000, and the learning rate is 0.001.

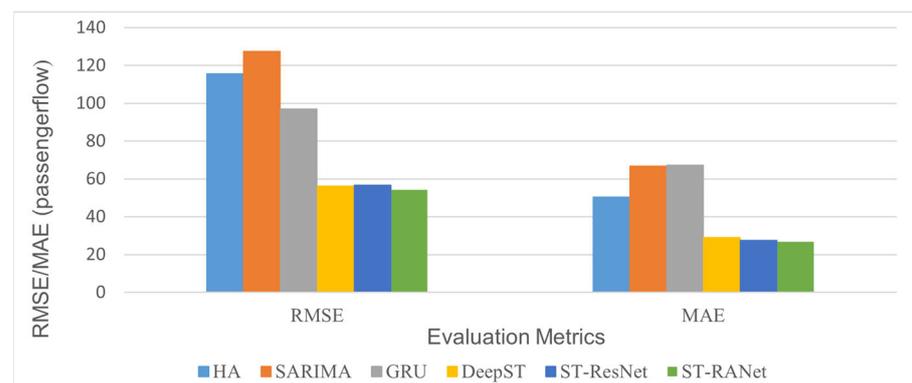
DeepST: A spatio-temporal data prediction model using CNN, which performs well in human flow predicting problems. The parameter setting is the same as ST-RANet.

ST-ResNet: An excellent model for predicting human traffic flow in space and time, with the same parameter settings as ST-RANet.

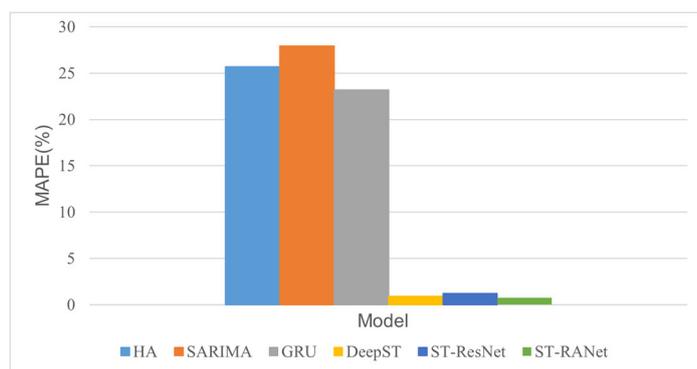
Table 3 and Figure 11 show the prediction performances of our ST-RANet model and other baseline models on SubwayBJ. Compared with the three time series models: HA, SARIMA, and GRU, it reduced RMSE and MAE by more than 78% and 87%, respectively. The MAPE indicator shows that less than 1% absolute error proportion is occupied, which reduces by 22–27%. Compared with the spatio-temporal model DeepST and ST-ResNet, our model also shows significant improvements in all metrics.

Table 3. Comparisons among different models on SubwayBJ.

Model	RMSE	MAE	MAPE(%)
HA [8]	115.56	50.40	25.70
SARIMA [10]	127.33	66.63	27.93
GRU [23]	96.82	67.17	23.20
DeepST [29]	56.04	29.01	0.91
ST-ResNet [31]	56.60	27.37	1.24
ST-RANet	54.24	26.84	0.78



(a) RMSE/MAE



(b) MAPE

Figure 11. Comparisons among different models on SubwayBJ.

As seen from Tables 2 and 3, high prediction results have been achieved with relatively small parameter values. With the resource allocation described above, the model training time is about 6000 s, which is less than one hour. The prediction time is only 0.3 s. This indicates that our model has low complexity and high expressiveness.

5.2. Ablation Experiments

The prediction results of our ST-RANet model for the outbound passenger flow of a station in the SubwayBJ dataset are shown in Figure 10.

In order to assess the impact of the CBAM module and external module of holiday and meteorological data on the model's performance, we conducted ablation experiments. We compared our ST-RANet with ST-ResNet without the CBAM module, ST-RANetExt without the external module and ST-ResNetExt without both CBAM and external modules. The metrics obtained from the ablation experiments are presented in Table 4.

Table 4. Results of ablation experiments on SubwayBJ.

Model	RMSE	MAE	MAPE(%)
ST-ResNet	56.60	27.37	1.24
ST-RANetExt	54.99	27.45	1.02
ST-ResNetExt	57.41	27.75	1.16
ST-RANet	54.24	26.84	0.78

It can be seen that firstly, with regard to the CBAM attention mechanism, our ST-RANet model has improved the prediction accuracy in every aspect compared to ST-ResNet. Secondly, in the ablation experiment of the external module, both ST-RANet and ST-ResNet models have shown significant improvement in prediction accuracy, indicating that holidays and meteorological factors contribute to the prediction of rail transit passenger flow.

5.3. Model Applicability

Since the SubwayBJ dataset only contains the flow data of Beijing Metro Line 13, to prove the good applicability of our ST-RANet model, we additionally processed the data of Beijing Metro Line 15 for the experiments. A 4×5 feature map was created for a total of 20 stations, and other settings are the same as the SubwayBJ dataset. The prediction results of the inbound traffic at a station of Line 15 are shown in Figure 12. The predictive indicators obtained by the model are presented in Table 5.

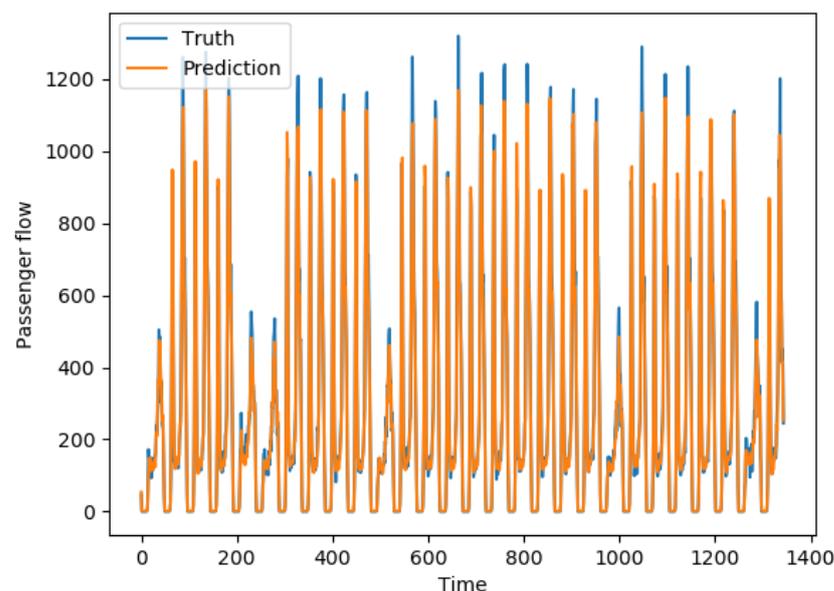


Figure 12. Prediction results of ST-RANet on Line 15.

Table 5. Predictive indicators on Beijing Metro Line 15.

	RMSE	MAE	MAPE(%)
ST-RANet-Line 15	35.47	18.58	0.62

Line 15 has an increase in network dimensions compared to the data in line 13. However, based on the results, it can be seen that on the dataset of Line 15, our ST-RANet model has a similar or even better prediction performance compared to Line 13, with a mean absolute error of less than 1% of the real data. With an accuracy of over 99%, it proves that our model has good generalization ability and applicability. When the model size increases, we can also adjust epochs, batch_size, and other hyperparameters in Table 2 to control the convergence speed and overfitting degree of model training, making the model have good scalability. In the future, the model can be applied to the entire Beijing metro network to predict the passenger flow of the whole network simultaneously.

6. Conclusions

In this paper, we introduce a deep learning-based spatio-temporal data prediction model that enables inbound and outbound passenger flow prediction for urban rail transit networks. Firstly, the spatio-temporal data is divided into three timescales: closeness, periodicity, and trend. Then the attention mechanisms are introduced between the residual units and the convolutional networks to process spatio-temporal passenger flow information. Moreover, external factors including holiday and meteorological data are fused to achieve more accurate predictions.

We evaluated our model using real data from several Beijing subway lines. Comparison with five baseline models using multiple evaluation metrics shows that the proposed ST-RANet model shows the best results. The ablation experiments demonstrate that taking external factors into account can cope with real data variations, and the model incorporating the attention mechanism can focus more on the parts that are helpful for the prediction goal, improving the model's modeling ability for time-series data. The experimental results show that our model has a prediction accuracy of more than 99%, which is highly applicable and significantly better than the other five baseline models.

In addition, there are certain limitations to our model. Firstly, it does not integrate other important factors that affect changes in passenger flow, such as major events. Secondly, the interpretability of the model is poor, as it is unable to show the entire process of data changes in the model. In the future, we will attempt to address these limitations and improve the applicability of the prediction model to fully meet the needs of actual operating work. Overall, passenger flow prediction technology has extensive application prospects and research significance, and our research will continue to focus on this technology to promote its better service to urban development and public welfare.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design, J.Y. and X.D.; data collection: H.Y.; analysis and interpretation of results: Y.W. and J.C.; draft manuscript preparation: X.D. and X.H.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Beijing Natural Science Foundation (grant No. L201015), and the National Special Project of Science and Technology Basic Resources Survey (grant No. 2022FY101400).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Readers can access the data used in this study by following this link: https://github.com/DXueRu/RANet_dataset.git, accessed on 11 September 2023.

Acknowledgments: We would sincerely want to thank the people who supported this work and the reviewing committee for their estimable feedback.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, X.; Li, S.; Tang, T.; Yang, L. Event-Triggered Predictive Control for Automatic Train Regulation and Passenger Flow in Metro Rail Systems. *IEEE Trans. Intell. Transport. Syst.* **2022**, *23*, 1782–1795. [[CrossRef](#)]
2. Luo, J.; Tong, Y.; Cavone, G.; Dotoli, M. A Service-Oriented Metro Traffic Regulation Method for Improving Operation Performance. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3533–3538.
3. Cavone, G.; Blenkins, L.; Van Den Boom, T.; Dotoli, M.; Seatzu, C.; De Schutter, B. Railway Disruption: A Bi-Level Rescheduling Algorithm. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23–26 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 54–59.
4. Xi, W.; Hu, M.; Wang, H.; Dong, H.; Ying, Z. Formation Control for Virtual Coupling Trains With Parametric Uncertainty and Unknown Disturbances. *IEEE Trans. Circuits Syst. II* **2023**, *70*, 3429–3433. [[CrossRef](#)]
5. Cavone, G.; Montaruli, V.; Van Den Boom, T.J.J.; Dotoli, M. Demand-Oriented Rescheduling of Railway Traffic in Case of Delays. In Proceedings of the 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT), Prague, Czech Republic, 29 June–2 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1040–1045.
6. Zhu, L.; Shen, C.; Wang, X.; Liang, H.; Wang, H.; Tang, T. A Learning Based Intelligent Train Regulation Method With Dynamic Prediction for the Metro Passenger Flow. *IEEE Trans. Intell. Transport. Syst.* **2023**, *24*, 3935–3948. [[CrossRef](#)]
7. Vlahogianni, E.I.; Golias, J.C.; Karlaftis, M.G. Short-term Traffic Forecasting: Overview of Objectives and Methods. *Transp. Rev.* **2004**, *24*, 533–557. [[CrossRef](#)]
8. El Esawey, M. Daily Bicycle Traffic Volume Estimation: Comparison of Historical Average and Count Models. *J. Urban Plann. Dev.* **2018**, *144*, 04018011. [[CrossRef](#)]
9. Williams, B.M.; Hoel, L.A. Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. *J. Transp. Eng.* **2003**, *129*, 664–672. [[CrossRef](#)]
10. Alharbi, F.R.; Csala, D. A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach. *Inventions* **2022**, *7*, 94. [[CrossRef](#)]
11. Liu, Z.; Liu, Y.; Meng, Q.; Cheng, Q. A Tailored Machine Learning Approach for Urban Transport Network Flow Estimation. *Transp. Res. Part C: Emerg. Technol.* **2019**, *108*, 130–150. [[CrossRef](#)]
12. Wang, P.; Liu, Y. Network Traffic Prediction Based on Improved BP Wavelet Neural Network. In Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, 12–17 October 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–5.
13. Su, H.; Yu, S. Hybrid GA Based Online Support Vector Machine Model for Short-Term Traffic Flow Forecasting. In *Advanced Parallel Processing Technologies*; Xu, M., Zhan, Y., Cao, J., Liu, Y., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4847, pp. 743–752; ISBN 978-3-540-76836-4.
14. Zarei, N.; Ghayour, M.A.; Hashemi, S. Road Traffic Prediction Using Context-Aware Random Forest Based on Volatility Nature of Traffic Flows. In *Intelligent Information and Database Systems*; Selamat, A., Nguyen, N.T., Haron, H., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7802, pp. 196–205; ISBN 978-3-642-36545-4.
15. Liao, Z.; Lan, P.; Fan, X.; Kelly, B.; Innes, A.; Liao, Z. SIRVD-DL: A COVID-19 Deep Learning Prediction Model Based on Time-Dependent SIRVD. *Comput. Biol. Med.* **2021**, *138*, 104868. [[CrossRef](#)]
16. Zhu, M.; Guan, X.; Li, Z.; He, L.; Wang, Z.; Cai, K. sEMG-Based Lower Limb Motion Prediction Using CNN-LSTM with Improved PCA Optimization Algorithm. *J. Bionic. Eng.* **2023**, *20*, 612–627. [[CrossRef](#)]
17. Polson, N.G.; Sokolov, V.O. Deep Learning for Short-Term Traffic Flow Prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17. [[CrossRef](#)]
18. Zhang, J.; Chen, F.; Cui, Z.; Guo, Y.; Zhu, Y. Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit. *IEEE Trans. Intell. Transport. Syst.* **2021**, *22*, 7004–7014. [[CrossRef](#)]
19. Kashyap, A.A.; Raviraj, S.; Devarakonda, A.; Nayak, K.S.R.; Santhosh, K.V.; Bhat, S.J. Traffic Flow Prediction Models—A Review of Deep Learning Techniques. *Cogent Eng.* **2022**, *9*, 2010510. [[CrossRef](#)]
20. Belhadi, A.; Djenouri, Y.; Djenouri, D.; Lin, J.C.-W. A Recurrent Neural Network for Urban Long-Term Traffic Flow Forecasting. *Appl. Intell.* **2020**, *50*, 3252–3265. [[CrossRef](#)]
21. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. [[CrossRef](#)]
22. Yang, D.; Chen, K.; Yang, M.; Zhao, X. Urban Rail Transit Passenger Flow Forecast Based on LSTM with Enhanced Long-term Features. *IET Intell. Transp. Syst.* **2019**, *13*, 1475–1482. [[CrossRef](#)]
23. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
24. Yang, X.; Xue, Q.; Yang, X.; Yin, H.; Qu, Y.; Li, X.; Wu, J. A Novel Prediction Model for the Inbound Passenger Flow of Urban Rail Transit. *Inf. Sci.* **2021**, *566*, 347–363. [[CrossRef](#)]
25. Dai, G.; Ma, C.; Xu, X. Short-Term Traffic Flow Prediction Method for Urban Road Sections Based on Space-Time Analysis and GRU. *IEEE Access* **2019**, *7*, 143025–143035. [[CrossRef](#)]
26. Zhang, J.; Chen, F.; Shen, Q. Cluster-Based LSTM Network for Short-Term Passenger Flow Forecasting in Urban Rail Transit. *IEEE Access* **2019**, *7*, 147653–147671. [[CrossRef](#)]

27. Wu, Y.; Tan, H.; Qin, L.; Ran, B.; Jiang, Z. A Hybrid Deep Learning Based Traffic Flow Prediction Method and Its Understanding. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 166–180. [[CrossRef](#)]
28. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors* **2017**, *17*, 818. [[CrossRef](#)] [[PubMed](#)]
29. Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X. DNN-Based Prediction Model for Spatio-Temporal Data. In Proceedings of the Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Burlingame, CA, USA, 31 October–3 November 2016; ACM: New York, NY, USA, 2016; pp. 1–4.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778.
31. Zhang, J.; Zheng, Y. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In Proceedings of the The Thirty-First AAAI Conference on Artificial Intelligence. The Twenty-Ninth Innovative Applications of Artificial Intelligence Conference. The Seventh Symposium on Educational Advances in Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31. [[CrossRef](#)]
32. Yang, J.; Dong, X.; Jin, S. Metro Passenger Flow Prediction Model Using Attention-Based Neural Network. *IEEE Access* **2020**, *8*, 30953–30959. [[CrossRef](#)]
33. Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention Augmented Convolutional Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seattle, WA, USA, 13–19 June 2020.
34. Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; Cottrell, G. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *arXiv* **2017**, arXiv:1704.02971.
35. Yang, S.; Liu, J.; Zhao, K. Space Meets Time: Local Spacetime Neural Network For Traffic Flow Forecasting. In Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM), Auckland, New Zealand, 7–10 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 817–826.
36. Fukushima, K. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biol. Cybern.* **1980**, *36*, 193–202. [[CrossRef](#)] [[PubMed](#)]
37. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11211, pp. 3–19; ISBN 978-3-030-01233-5.
38. Takekawa, A.; Kajitara, M.; Fukuda, H. Role of Layers and Neurons in Deep Learning With the Rectified Linear Unit. *Cureus* **2021**, *13*, 18866. [[CrossRef](#)] [[PubMed](#)]
39. Zheng, Y. Methodologies for Cross-Domain Data Fusion: An Overview. *IEEE Trans. Big Data* **2015**, *1*, 16–34. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.