Deep-learning Architecture for Short-term Passenger Flow Forecasting in Urban Rail Transit

Jinlei Zhang, Feng Chen, Yadi Zhu, and Yinan Guo

Abstract—Short-term passenger flow forecasting is an essential component in urban rail transit operation. Emerging deep-learning models provide good insight into improving prediction precision. Therefore, we propose a deep-learning architecture combined residual network (ResNet), graph convolutional network (GCN) and long short-term memory (LSTM) (called “ResLSTM”) to forecast short-term passenger flow in urban rail transit on a network scale. First, improved methodologies of ResNet, GCN, and attention LSTM models are presented. Then, model architecture is proposed, wherein ResNet is used to capture deep abstract spatial correlations between subway stations, GCN is applied to extract network-topology information, and attention LSTM is used to extract temporal correlations. Model architecture includes four branches for inflow, outflow, graph-network topology, as well as weather conditions and air quality. To the best of our knowledge, this is the first time short-term passenger flow forecasting (STPFF) can provide real-time passenger flow information for passengers to make rational scheduling decisions and for transit operators to control ridership inflow, thereby avoiding congestion, or to adjust train timetables, thereby being able to accommodate more passengers in peak hours. Many researchers have devoted considerable effort to studying STPFF in URT. Consequently, STPFF has developed in three stages in recent decades.

The initial stage is represented by conventional mathematical statistics-based methods such as historical average [1], ordinary least squares [2], logistic regression [3], autoregressive integrated moving average (ARIMA) [4, 7], Kalman filter [8, 9], and k-nearest neighbor [10] models. Many researchers have also summarized these models [11, 12] because initially, URT did not develop so fast that researchers neglected to apply STPFF to URT. However, most of these models are no longer used to analyze road traffic because they cannot meet “real-time” requirements and cannot achieve higher precision than state-of-the-art models.

The second stage is represented by machine-learning and hybrid prediction models such as backpropagation neural networks (BPNNs) [13], decision-tree learning [14], random-forest learning [15], and support-vector machine (SVM) [16] models. In this period, more studies began to focus on STPFF in URT with its gradual development. For example, Roos et al. [17] built a dynamic Bayesian network to conduct STPFF in URT and combined dynamic Bayesian networks with Gaussian mixture models [18]. Li et al. [19] built multiscale radial-basis-function networks to perform STPFF in URT. Wang et al. separately combined ARIMA with both wavelet decomposition and SVM to conduct STPFF in URT [20, 21]. Yu et al. combined empirical mode decomposition and BPNN to carry out STPFF in URT. These hybrid models exhibit significantly better prediction precision than most mathematical statistics-based models. Moreover, it has been demonstrated that hybrid models often outperform corresponding single ones. However, most of the foregoing models cannot consider spatial correlations in model formulation, which is critically important to improve prediction precision. Furthermore, researchers always take one or several subway stations as examples to verify their models. For network STPFF, e.g., several hundred subway stations in a network, such models cannot perform well.

The third stage is represented by deep-learning models. Since deep-belief networks were initially proposed in 2006 [22], deep neural networks (DNNs) have received enormous attention for application to computer vision and natural
language processing. With rapid expansion of URT, numerous researchers have applied DNNs to achieve STPFF in URT. In the last several years in particular, many studies on use of models such as long short-term memory (LSTM) [23], gated recurrent units (GRUs) [24], convolutional neural networks (CNNs) [25], convolutional LSTM (ConvLSTM) [26], [27], graph convolutional networks (GCNs), stacked autoencoders (SAEs) [28], and many variants thereof to achieve STPFF in URT and road traffic have been published. Although Guo et al. [29] proposed a hybrid SVM–LSTM model to predict short-term abnormal flow in URT, they did not take spatial correlations between stations into account. Tang et al. [30] proposed a spatiotemporal LSTM model, wherein time-cost and spatial-correlation matrices were used to extract spatial correlations. This study only involved outflow. However, the inflow is more significant for operation. Although Liu et al. [31] built a deep-learning-based architecture by considering spatiotemporal, environmental, and operational factors, their model was so complicated, it could be applied only to several subway stations rather than a whole network. Li et al. [32] introduced a graph CNN to conduct STPFF in URT, wherein manually constructed historical origin-destination (OD) matrices were used as inputs. However, they neglected that a real-time OD matrix could not be obtained [33]. Similarly, Han et al. [34] built spatiotemporal graph CNNs to predict short-term ridership in a citywide metro network. Although they used network topology to extract spatial correlations between stations, they did not consider external factors such as weather conditions, events, or air quality. Since residual neural networks (ResNets) have been proposed [35], some researchers have applied them to road-traffic models of bus [36] and taxi [37] flow predictions and have proved its ability in STPFF. To the best of our knowledge, however, there have been only a few such applications of ResNet to STPFF in URT. In this stage, the STPFF has greatly improved prediction precision and feasibility on a network scale. However, few researchers have considered factors such as air quality, which greatly influence passenger travel.

On the basis of shortcomings of previous studies, we build a deep-learning architecture called “ResLSTM,” combing ResNet, GCN, and attention LSTM to conduct STPFF in URT on a network scale. In addition to spatiotemporal correlations between subway stations, topological relationships between them, as well as weather conditions and air quality are also incorporated into ResLSTM to determine how such factors affected passenger travel. The main contributions of the proposed architecture are as follows.

1) The proposed ResLSTM model considers not only spatiotemporal features but also network topology as well as weather conditions and air quality. Moreover, making real-time predictions with high prediction precision on a network scale is realized.

2) The proposed ResLSTM architecture is so robust that the effect of even deleting one of the four branches on prediction precision is negligible.

3) Influences of weather conditions and air quality on prediction accuracy are quantified.

4) We compare performance of the proposed ResLSTM model with those of several prevailing state-of-the-art models, and the proposed ResLSTM model outperforms all of them. Furthermore, comparison of prediction precisions obtained from time granularities of 10, 15, and 30 min indicates that prediction precision increases with increasing time granularity (TG).

The remainder of this paper is organized as follows. In section II, methodologies of ResNet, GCN, and attention LSTM are presented. In section III, architecture of the proposed ResLSTM model including input and output data are described. In section IV, case study results are discussed and compared with results obtained using several prevailing state-of-the-art models. The main findings and limitations of the current study and their significance are summarized and a direction for future research is proposed in section V.

II. METHODOLOGY

Architecture of our model was designed mainly based on ResNet, GCN, and attention LSTM. Therefore, we briefly introduce their respective methodologies herein. In every subsection, architecture used in our model is presented.

A. ResNet

Network passenger flow can be treated as preprocessed images [25]. Therefore, we can use CNN to extract low-, mid-, and high-level abstract spatiotemporal features of passenger flow [38]. Previous studies have shown that deeper models can extract more enriched features, thus achieving better performance [39], [40]. However, deeper models are not necessarily better ones because of vanishing or exploding gradients during model training [41]. Hence, He et al. [35] proposed ResNet containing a skip connection in 2015, as shown in Fig. 1 (top), and achieved favorable results. The purpose of the model is to train network output, as shown in (1).

\[ X_{t+1} = F(X_t) + X_t, \]

where \( X_t \) and \( X_{t+1} \) represent residual block input and output, respectively.

In this study, we adopt an improved residual block, also previously proposed by He et al. [42], as shown in Fig. 1 (bottom). The difference between the two versions is that in the improved residual block, gradients can be passed unimpededly to any earlier layers through shortcut connections, thus solving the vanishing or exploding gradient problem when training deeper networks. An example of a residual block showing 32 filters, as used in our study, is shown in Fig. 2. “Conv” indicates a convolutional layer. “BN” denotes a batch-normalization layer. “ReLU” represents an activation layer. To ensure input and output show the same dimensions so that the former can be added to the latter, we added a convolutional layer before the residual block.
is the identity matrix. $W$ is the weight matrix of the $l$th neural network layer, and $\sigma(\cdot)$ is a nonlinear activation function such as ReLU.

$$H^{l+1} = f(H^l, A) = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^lW^l),$$

$$H' = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H,$$

where $\tilde{A} = A + I$, $\tilde{D}$ is the diagonal node-degree matrix of $\tilde{A}$, $I$ is the identity matrix. $W$ is the weight matrix of the $l$th neural network layer, and $\sigma(\cdot)$ is a nonlinear activation function such as ReLU.

Fig. 3. Graph convolutional network.

However, some previous works have shown that stacking multiple GCN layers leads to not only higher complexity during backpropagation but also gradient vanishing [44, 45], thereby degrading performance of deeper GCNs [43, 46, 47]. Moreover, oversmoothing, wherein multiple features of the same vertex converge to the same value, is also a common problem that arises with deeper GCNs [48]. Therefore, in this study, we extend GCN to ResNet GCN to mitigate such drawbacks. Let $In \in R^{s \times s}$ be input, where $s$ is the number of subway stations, and $t$ represents historical timesteps for each station. Each input matrix is treated as a graph signal, which then is transformed according to (4).

$$In' = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}In,$$

where $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is the symmetric normalized Laplacian, as shown in (3). Transformed $In'$ has the same shape as $In$ and contains rich network topological information, which was subsequently used as ResNet input.

C. Attention LSTM

LSTM has been shown effective in predicting traffic flow [23, 24, 30, 49, 50]. Because the attention mechanism has been used successfully for machine translation since its introduction [51], many researchers have applied it to STPFF [52 – 54]. Therefore, to capture different weights of features extracted from former network layers, we introduced attention LSTM to our model.

Conventional attention LSTM is used to capture weight scores of different timesteps, usually by assigning heavier weight scores to adjacent timesteps and lower ones to those further apart. However, traffic prediction models, which are affected by many factors such as weather conditions, passenger enter and exit flows, and network topology, are so sophisticated that assigning weight scores based on recentness is insufficient [54]. Therefore, on the basis of previous work by Wu et al. [54], we use a fully connected network to obtain weights, which can be scored according to input or LSTM output. Test results indicated that the latter was more effective; therefore, LSTM output weight was automatically scored. Let matrix $Out \in R^{m \times n}$ be LSTM output, where $m$ and $n$ represent timesteps and number of features of each timestep, respectively.
attention-based output ($Out'$) can be obtained by (5) and (6).

\[
A = f(W \odot Out + b), \quad (5)
\]
\[
Out' = A \odot Out, \quad (6)
\]

where $A$ is a weight matrix whose shape is identical to that of $Out$, “$\odot$” denotes Hadamard product, “$f$” represents the fully connected layer (which can be activated by different activation functions such as sigmoid functions), $W$ is the weight matrix of $f$, and $b$ is bias.

III. MODEL DEVELOPMENT

Herein, we describe ResLSTM model architecture, as shown in Fig. 4, which comprises four branches. All input data are obtained for times $t-n$ to $t$, and output data are inflows from time $t+1$. Branch 1 uses inflows to capture spatiotemporal features. Branch 2 is identical to branch 1 except it used outflows. Branch 3 extracts network topological information. Branch 4 represents impacts of weather conditions and air quality on prediction precision. Moreover, attention LSTM is used in the trunk to obtain output data. Detailed model architecture description is presented in subsections A to F.

![ResLSTM model architecture](image)

**A. Branch 1: Inflow**

For predicting network inflow, prior knowledge of historical inflow is the most important criterion. Therefore, branch 1 captured inflow. In previous studies, the relationship between inflow and outflow always has been indicated by a model showing two channels [32], [34], [37], one corresponding to inflow; the other, outflow. However, when three patterns (such as real-time, daily, and weekly patterns) must be considered, the model should have three branches, which significantly increases complexity; therefore, we chose an ingenious method of separately considering inflow and outflow. Initially, for the same station in a subway-station network, inflow and outflow were only slightly related, which is an initial condition completely different from that for a road-traffic network. A subway station showing heavy inflow may show light outflow. Furthermore, treating inflow and outflow separately can decrease model complexity without decreasing prediction precision.
Nowadays, real-time inflow and outflow in a subway station can be obtained by automatic fare collection (AFC); therefore, we considered three inflow patterns; namely, real-time, daily, and weekly patterns. The inflow time series is shown in (7).

\[
X_{s,t}^P = \begin{pmatrix}
  x_{1,t-n}^p & x_{1,t-n+1}^p & x_{1,t-n+2}^p & \cdots & x_{1,t}^p \\
  x_{2,t-n}^p & x_{2,t-n+1}^p & x_{2,t-n+2}^p & \cdots & x_{2,t}^p \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  x_{s,t-n}^p & x_{s,t-n+1}^p & x_{s,t-n+2}^p & \cdots & x_{s,t}^p
\end{pmatrix}, \quad (7)
\]

where \( s \) is the number of subway stations, and \( t \) represents historical timesteps for each station. Stations are ordered according to their line number, e.g., line 1, line 2, etc. In each line, adjacent stations are placed in adjacent rows according to train direction, where “\( p \)" represents different patterns. If \( p \) represents real-time, daily, or weekly patterns, \( X_{s,t}^r \) and \( X_{s,t}^w \) represent inflow time series corresponding to the same day, previous day, or previous week, respectively.

As shown in branch 1, to predict inflow for time \( t + 1 \), we organize time series from three patterns for times \( t-4 \) to \( t \) into a single three-channel “image.” Branch 1 input is shown in (8).

\[
I_1 = (X_{s,t}^r, X_{s,t}^d, X_{s,t}^w). \quad (8)
\]

Data are input into two residual blocks, the first showing 32 filters; the second, 64. Then, data are flattened and fully connected with 276 neurons. Branch 1 output data are then input into the feature-fusion section.

B. Branch 2: Outflow

Outflow processing for branch 2 is identical to inflow processing for branch 1. Hence, branch 2 input is shown in (9).

\[
I_2 = (X_{s,t}^r, X_{s,t}^d, X_{s,t}^w). \quad (9)
\]

where \( X' \) represents outflow.

C. Branch 3: Graph signal

Traffic-network topology has been proven important for STPFF [32], [34]. To overcome some drawbacks stated in section II-B, we use a ResNet GCN, as shown in branch 3, to capture influence of network topology. We only considered the real-time pattern because network topology did not change. According to (4), (7), and (8), original input for ResNet GCN is shown in (10). Input data are subsequently processed according to the identical method described for branch 1.

\[
I_3 = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} (X_{s,t}'). \quad (10)
\]

D. Branch 4: Weather conditions and air quality

Although only a few researchers have considered impact of weather conditions on STPFF [31]. To the best of our knowledge, none have considered impact of air quality on STPFF. However, weather conditions and air quality are both essential information for people scheduling their travel plans. For example, cold weather and heavily polluted air often impede passengers’ nonemergency, nonessential travels.

The dataset contained two subsets: weather conditions showing real-time temperature (°C), dew point temperature (°C), relative humidity (%), and wind speed (m/s), all of which were recorded every half-hour, as shown in TABLE I; and air quality showing real-time air-quality index (AQI) and concentrations of atmospheric particulate matter (PM2.5 and PM10), SO2, NO2, CO, and O3 (μg/m3), all of which were recorded every hour, as shown in TABLE II.

TABLE I

| Date/Time | Temperature | Dew point temperature | Relative humidity | Wind speed |
|-----------|-------------|-----------------------|------------------|------------|
| 2016/2/29 5:00 | -6 | -17 | 42 | 4 |
| 2016/2/29 5:30 | -6 | -17 | 42 | 4 |
| 2016/2/29 6:00 | -5 | -17 | 39 | 11 |
| 2016/2/29 6:30 | -5 | -18 | 36 | 7 |
| 2016/2/29 7:00 | -6 | -18 | 39 | 4 |
| 2016/2/29 7:30 | -6 | -16 | 46 | 7 |
| 2016/2/29 8:00 | -3 | -16 | 37 | 4 |

TABLE II

| Date/Time | AQI | PM2.5 | PM10 | SO2 | NO2 | CO | O3 |
|-----------|-----|-------|------|-----|-----|----|----|
| 2016/2/29 5:00 | 18 | 11 | 16 | 5 | 36 | 0.5 | 38 |
| 2016/2/29 6:00 | 21 | 13 | 21 | 6 | 40 | 0.5 | 37 |
| 2016/2/29 7:00 | 20 | 14 | 20 | 5 | 38 | 0.5 | 40 |
| 2016/2/29 8:00 | 20 | 12 | 20 | 5 | 37 | 0.5 | 41 |
| 2016/2/29 9:00 | 24 | 15 | 24 | 6 | 35 | 0.5 | 47 |
| 2016/2/29 10:00 | 25 | 17 | 24 | 7 | 33 | 0.6 | 53 |
| 2016/2/29 11:00 | 28 | 19 | 26 | 7 | 33 | 0.6 | 54 |

If we conduct STPFF at TG = 10 min, weather-condition data from 05:00 to 05:10 will share recorded data from 05:00 to 05:30, as shown in the first row of TABLE I. Similarly, corresponding air-quality data from 05:00 to 05:10 will share the recorded data from 05:00 to 06:00, as shown in the first row of TABLE II. We obtain preprocessed input data for branch 4, as shown in (11).

\[
I_4 = X_{w,t} = \begin{pmatrix}
  x_{1,t-n} & x_{1,t-n+1} & x_{1,t-n+2} & \cdots & x_{1,t} \\
  x_{2,t-n} & x_{2,t-n+1} & x_{2,t-n+2} & \cdots & x_{2,t} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  x_{s,t-n} & x_{s,t-n+1} & x_{s,t-n+2} & \cdots & x_{s,t}
\end{pmatrix}. \quad (11)
\]

where \( w \) represents 11 indicators for weather-condition and air-quality data.

Preprocessed input data are flattened and subsequently added to the fully connected layer to obtain weighted indicators. Stacked LSTM showing 128 and 276 neurons for the first and second layers, respectively, is then applied. Output data are then input into the feature-fusion section.

E. Feature fusion

Because data output from the four branches shows identical shape, weighted feature-fusion is easily implemented according to (12).
Fusion = \( W_1 \odot O_1 + W_2 \odot O_2 + W_4 \odot O_3 + W_4 \odot O_4 \)\), \(O_1, O_2, O_3\), and \(O_4\) are outputs from the four branches, \(W\) is the corresponding weight vector used to capture degrees of impact of different features, and “\(\odot\)” represents the Hadamard product.

Attention LSTM, as described in section II-C, is applied after feature fusion \([30]\). LSTM output subsequently was flattened and fully connected with 276 neurons to generate the final output.

F. Loss function and model training

We adopt end-to-end training to optimize the neural network. Mean-squared error (MSE), as shown in (13), is the loss function used in this study. The “Adam” optimizer is selected to train the model and optimize the network.

\[
\text{Loss} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{13}
\]

where \(y_i\) is the actual value, \(\hat{y}_i\) is the predicted value, and \(n\) is the number of samples.

We apply three indicators to evaluate model performance; namely, root-mean-squared error (RMSE), mean-absolute error (MAE), and weighted-mean-absolute-percentage error (WMAPE), as given by (14) – (16), respectively.

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}, \tag{14}
\]

\[
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{15}
\]

\[
\text{WMAPE} = \frac{\sum_{j=1}^{n} |y_j - \hat{y}_j| \sum_{j=1}^{n} y_j}{\sum_{j=1}^{n} y_j}, \tag{16}
\]

where \(\sum_{j=1}^{n} y_j\) is the sum of actual values.

IV. CASE STUDY RESULTS AND DISCUSSION

Herein, we describe the dataset, present detailed model configuration, and compare performances of the different models.

A. Dataset description

AFC data used in this study are collected from Beijing subway between 05:00 and 23:00 for five consecutive weeks from February 29 to April 3, 2016. There were 17 lines and 276 subway stations (excluding the airport express line and the stations on it) in March 2016 in Beijing. Only data (containing 130 million records) from the 25 workdays in that period are applied to this study. Every record contains card number, entry-station number, exit-station number, entry time, exit time, entry-station name, and exit-station name. Inflow and outflow time series are extracted according to (7). TGs used in our study are 10, 15, and 30 min. Therefore, it is easy to integrate results predicted for TG = 10 and 15 min into those predicted for TG = 30 min to compare prediction performances for different TGs. Examples of weather-condition and air-quality datasets are shown in section III-D.

B. Model configuration

We use data from the first four weeks to train models and data from the final week to test them. To achieve tradeoff between model-training time and prediction precision, we use the previous five network timesteps to forecast the next one by trial and error. For branch 1, the first residual block has 32 filters; the second, 64. The fully connected layer consists of 276 neurons. Branches 1, 2, and 3 show the same configuration. For branch 4, fully connected layers consist of 276 neurons; and the two LSTM layers, 128 and 276, respectively. For feature fusion, the attention LSTM and final fully connected layers consist of 128 and 276 neurons, respectively. MSE and the “Adam” are the loss function and optimizer, respectively. We train the models ten times to avoid randomness and obtain the best performance.

We compare performances of three kinds of models. First, we choose ARIMA, a representative conventional mathematical statistics-based model. We use Expert Modeler in Statistical-Package-for-the-Social-Sciences (SPSS®) software (IBM Corp., USA) to obtain ARIMA results. Second, we choose BPNN and support-vector regression (SVR) because they are representative machine-learning models. BPNN has two hidden layers, each containing 1000 neurons. The SVR kernel is a radial-basis function (RBF–SVR). Inputs are inflows of the previous thirty timesteps obtained from real-time patterns for the 276 subway stations. Outputs are inflows of the next timestep for the same. Third, CNN, LSTM, GRU, CNN, ConvLSTM, and ResLSTM-GCN, all of which have two kernel layers, are chosen for deep-learning models. The two kernel layers of CNN, LSTM, and GRU, each have 1000 neurons, and inputs and outputs are identical to those for BPNN. For both CNN and ConvLSTM, the first and second kernel layers has 32 and 64 filters, respectively. Inputs are both inflows and outflows of the previous five timesteps from the three patterns obtained for the 276 subway stations. Outputs are inflows of the next timestep for the same. ResLSTM-GCN is the same as branch 3, as stated in sections III-B and III-C. Moreover, when using ARIMA to conduct STPFF on a subway network, we have to build 276 models representing each of the 276 stations. Except for ARIMA, all the benchmark models are used to obtain overall results for the 276 stations by training only a single model. All the models are trained and tuned to obtain the best performances.

One important contribution of the proposed ResLSTM model is that it can determine impacts of different branches on prediction performance. Therefore, to determine influences of network topology as well as weather conditions and air quality (e.g., PM2.5 and PM10) on model-prediction performance, we simply deleted branches 3 and 4 (i.e., “ResLSTM-No graph” and “ResLSTM-No W&A,” respectively) and conducted the same prediction experiments.

As stated in section III-A, the proposed ResLSTM model is advanced in that flows can be organized into three channels rather than two. To verify model performance, we organized inflow and outflow as two channels (i.e., “ResLSTM-TC”). That is, branches 1 and 2 containing weekly, daily, and real-time inflows and outflows, respectively, are subsequently...
transformed into three branches, each containing one pattern and two channels (i.e., weekly inflow and outflow, daily inflow and outflow, and real-time inflow and outflow). Other model configurations do not change among the three ResLSTM variant models.

C. Results and discussion

Prediction performances obtained for different models are shown in TABLE IV and Fig. 5. As shown in TABLE IV, deep-learning-based models greatly outperform mathematical statistics-based and machine-learning-based ones in most cases. RBF–SVR performs the worst. SVR is unsuitable for regression of large datasets owing to its higher computation cost. The second-worst model is ARIMA. Although each subway station has its own individual model in ARIMA, prediction performance is also poor because ARIMA cannot capture comprehensive nonlinear features of passenger flows.

Among deep-learning-based models, all the convolution-based models perform better than recurrence-based ones when only a single model is used for network predictions. LSTM and GRU perform better than RNN, as expected. ConvLSTM performs better than CNN because ConvLSTM can capture more temporal information than CNN. However, ConvLSTM performance worsens with increasing TG. Decrease in number of samples may account for such phenomena. Among ResLSTM and its numerous variants, ResLSTM performs best because many features including inflow, outflow, network topology, as well as weather conditions and air quality are taken into account. It is worth mentioning that the proposed architecture presented strong robustness; that is, even when one branch is deleted, prediction results do not change significantly (see results of ResLSTM-No Graph and ResLSTM-No W&A). Moreover, satisfactory results are obtained using only branch 3 (see results of ResLSTM-GCN), which strongly demonstrates robustness of the proposed architecture. Although all the models show similar prediction performances when TG = 10 min, ResLSTM begins to show its superior prediction performance and gaps between performances of ResLSTM and its variants widen when TG is increased from 10 to 30 min. Moreover, ResLSTM always performs best regardless of whether TG = 10, 15, or 30 min.

Comparing prediction performances of ResLSTM-No Graph and ResLSTM shows that network topology has some influence on prediction precision. Comparing prediction performances of ResLSTM-No W&A and ResLSTM, we can infer using common sense that introduction of weather-condition and air-quality datasets increases prediction precision. If weather is very cold or air pollution is critically high (e.g., higher than PM2.5 and PM10), people will reduce or eliminate unnecessary travel, meaning these external factors will affect passenger volume. Moreover, the influence is quantified, with RMSE, MAE, and WMAPE all decreasing from 60.1340 to 56.9649, 34.1360 to 32.5819, and 6.428 to 6.134%, respectively, when TG = 30 min. Comparing prediction performances of ResLSTM-TC and ResLSTM, we can infer that treating passenger flow separately not only will save computation cost but also will not decrease prediction precision.

To compare prediction precisions obtained for different TGs, we aggregate results obtained when TG = 10 and 15 min into those obtained when TG = 30 min and then compute corresponding evaluation indicators. The results are shown in TABLE III. As demonstrated, prediction precision gradually increases with increasing TG. RMSE, MAE, and WMAPE all decreases from 61.5458 to 56.9649, 35.2239 to 32.5819, and 6.633 to 6.134%, respectively. From a statistical perspective, that is because the passenger-flow similarity and regularity with those of corresponding historical passenger flows will increase when flows are aggregated under larger TGs, thereby contributing to better overall prediction precision.

In summary, the proposed ResLSTM model presents satisfactory ability to conduct STPFF in URT. It also shows strong robustness, favorable for practical real-world application. When TG = 10 min, RMSE, MAE, and WMAPE are 28.7394, 17.0315, and 9.579%, respectively. When TG = 15 min, RMSE, MAE, and WMAPE are 36.0444, 20.8783, and 7.805%, respectively. When TG = 30 min, RMSE, MAE, and WMAPE are 56.9649, 32.5819, and 6.134%, respectively.
model branches and feature-fusion section were presented in detail. Finally, the model was applied to an actual subway system in Beijing, and its prediction performance was compared with those of many state-of-the-art models to show the advancement of our proposed model. To the best of our knowledge, this is the first time that air-quality indicators such as PM2.5 and PM10 have been considered in STPFF. Moreover, influences of weather conditions and air quality on prediction precision were quantified. In addition, we combined GCN and ResNet to overcome drawbacks of GCN. The main conclusions are summarized as follows.

1) The ResLSTM model could capture not only spatiotemporal features of passenger flows but also network topological information as well as influences of weather conditions and air quality on prediction precision.

2) The ResLSTM model shows strong robustness, essential for real-world application. Moreover, prediction precisions are favorable when conducting short-term passenger flow forecasting on a network scale for nearly 300 subway stations.

3) Weather conditions and air quality are proved to have considerable influence on prediction precision and the influence is quantified.

4) Prediction precision increases with increasing TG because of higher similarity and regularity when passenger flows are aggregated under a higher TG.

There are also some limitations to our study. For example, we did not consider weekend passenger flows owing to substantial fluctuations and less regularity. Multistep predictions can also be explored. Moreover, model interpretability is poor because such a model is exactly like a “black box”, wherein data are fed to obtain satisfactory predictions. Future studies can try to compensate for these limitations.

V. CONCLUSION

In this study, we first summarized development of STPFF according to mathematical statistics, machine-learning, and deep-learning stages and discussed disadvantages of various models in each stage. Then, methodologies of ResNet, GCN, and attention LSTM were described. Based on improved methodology, the ResLSTM model was developed. The four

| Category | TG Indicators | 10 min | 15 min | 30 min |
|----------|---------------|--------|--------|--------|
|          | RMSE          | MAE    | WMAPE  | RMSE   | MAE    | WMAPE  | RMSE   | MAE    | WMAPE  |
| MS       | ARIMA         | 50.5436| 27.3968| 15.53% | 79.9580| 42.2139| 15.95% | 189.3329| 100.3590| 18.95% |
| ML       | RBF-SVR       | 72.3753| 61.2418| 34.70% | 118.5395| 100.2861| 37.63% | 256.0734| 206.0826| 39.33% |
|          | BPNN          | 45.6946| 24.5322| 14.24% | 73.3083 | 40.8457| 15.45% | 163.9720| 87.1259 | 16.81% |
| DL       | RNN           | 49.6506| 26.2275| 14.75% | 74.6021 | 41.4159| 15.43% | 127.9259| 71.8328 | 13.84% |
|          | LSTM          | 37.1903| 21.9925| 12.71% | 53.9216 | 29.5340| 11.29% | 96.3534 | 55.8265 | 10.76% |
|          | GRU           | 44.9115| 24.7353| 14.33% | 56.6550 | 33.0309| 12.56% | 86.6984 | 52.0177 | 10.03% |
|          | CNN           | 29.8125| 18.5460| 10.413%| 40.2673 | 25.1231| 9.375% | 64.0458 | 39.6867 | 7.472% |
|          | ConvLSTM      | 28.7943| 17.4780| 9.814% | 37.0923 | 22.4236| 8.580% | 61.4978 | 36.9768 | 6.962% |
|          | ResLSTM-GCN   | 30.0706| 17.4331| 9.775% | 36.7801 | 21.6615| 8.099% | 59.7289 | 33.6304 | 6.334% |
|          | ResLSTM-No Graph | 28.9342| 17.4870| 9.827% | 36.4708 | 20.9008| 7.815% | 58.7114 | 33.2383 | 6.260% |
|          | ResLSTM-No W&A | 29.6219| 17.6344| 9.911% | 38.7386 | 23.0008| 8.589% | 60.1340 | 34.1360 | 6.428% |
|          | ResLSTM-TC    | 30.6557| 17.2621| 9.692% | 39.2743 | 23.8779| 8.901% | 70.0569 | 42.7349 | 8.046% |
|          | ResLSTM       | 28.3661| 16.6318| 9.352% | 36.0444 | 20.8783| 7.805% | 56.9649 | 32.5819 | 6.134% |

Note: MS is mathematical-statistics-based model. ML is machine-learning-based models. DL is deep-learning-based models.
REFERENCES

[1] Y.J.S. EDESPG. Michalopoulos and R.A. Plum, "Improved estimation of traffic flow for real-time control," and Characteristics, vol. 7, no. 9, pp. 28, 1980.

[2] H. Sun, H.X. Liu, H. Xiao, R.R. He and B. Ran, "Use of local linear regression model for short-term traffic forecasting," Transport. Res. Rec., vol. 1836, no. 1, pp. 143-150, 2003.

[3] M. Abdel-Aty, N. Uddin, A. Pande, M.F. Abdalla and L. Hsia, "Predicting freeway crashes from loop detector data by matched case-control logistic regression," Transport. Res. Rec., vol. 1897, no. 1, pp. 88-95, 2004.

[4] B.M. Williams and L.A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," J. Transp. Eng., vol. 129, no. 6, pp. 664-672, 2003.

[5] C. HanS. Song and C. Wang, "A real-time short-term traffic flow adaptive forecasting method based on ARIMA model [J]," Acta Simulata Systematica Sinica, vol. 7, pp. 43, 2004.

[6] X. Min, J. Hu, Q. Chen, T. Zhang and Y. Zhang, Short-term traffic flow forecasting of urban network based on static STARIMA model, Proc. 2009 12th International IEEE conference on intelligent transportation systems, IEEE, 2009, pp. 1-6.

[7] S. Shekhar and B.M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," Transport. Res. Rec., vol. 2024, no. 1, pp. 116-125, 2007.

[8] F. Yang, Z. Yin, H. Liu and B. Ran, "Online recursive algorithm for short-term traffic prediction," Transport. Res. Rec., vol. 1879, no. 1, pp. 1-8, 2004.

[9] I. Okutani and Y.J. Stephanes, "Dynamic prediction of traffic volume through Kalman filtering theory," Transportation Research Part B: Methodological, vol. 18, no. 1, pp. 1-11, 1984.

[10] G.A. Davis and N.L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," J. Transp. Eng., vol. 117, no. 2, pp. 178-188, 1991.

[11] B.L. Smith and M.J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," J. Transp. Eng., vol. 123, no. 4, pp. 261-266, 1997.

[12] E.I. Vlahogianni, C. Golas and M.G. Karlaftis, "Short-term traffic flow forecasting: Overview of objectives and methods," Transport reviews, vol. 24, no. 5, pp. 533-557, 2004.

[13] P. Wang and Y. Liu, Network traffic prediction based on improved BP wavelet neural network, Proc. 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2008, pp. 1-5.

[14] Y. XuQ. Kong and Y. Liu, Short-term traffic flow volume prediction using classification and regression trees, Proc. 2013 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2013, pp. 493-498.

[15] N. ZaremB.A. Ghayour and S. Hashemi, Road traffic prediction using context-aware random forest based on volatility nature of traffic flows, Proc. Asian Conference on Intelligent Information and Database Systems, Springer, 2013, pp. 196-205.

[16] H. Su and S. Yu, Hybrid GA based online support vector machine model for short-term traffic flow forecasting, Proc. International Workshop on Advanced Parallel Processing Technologies, Springer, 2007, pp. 743-752.

[17] J. RoosG. Gavin and S. Bonneya, "A dynamic Bayesian network approach to forecast short-term urban rail passenger flows with incomplete data," Transportation Research Procedia, vol. 26, pp. 53-61, 2017.

[18] J. RoosS. Bonney and G. Gavin, Dynamic Bayesian networks with Gaussian mixture models for short-term passenger flow forecasting, Proc. International Conference on Intelligent Systems and Knowledge Engineering, 2017, pp. 1-8.

[19] Y. Li, X. Wang, S. Sun, X. Ma and G. Lu, "Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks," Transportation Research Part C: Emerging Technologies, vol. 77, pp. 306-328, 2017.

[20] X. Wang, N. Zhang, Y. Chen and Y. Zhang, Short-term forecasting of urban rail transit ridership based on ARIMA and wavelet decomposition, Proc. AIP Conference Proceedings, AIP Publishing, 2018, pp. 40025.

[21] X. Wang, N. Zhang, Y. Zhang and Z. Shi, "Forecasting of Short-Term Metro Ridership with Support Vector Machine Online Model," J. Adv. Transport., vol. 2018, 2018.

[22] G.E. HintonS. Osindero and Y. Teh, "A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, no. 7, pp. 1527-1554, 2006.

[23] X. Ma, T. Tao, Y. Wang, H. Yu and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," Transportation Research Part C: Emerging Technologies, vol. 54, pp. 187-197, 2015.

[24] R. FuZ. Zhang and L. Li, Using LSTM and GRU neural network methods for traffic flow prediction, Proc. Chinese Association of Automation (YAC), Youth Academic Annual Conference of IEEE, 2016, pp. 324-331.

[25] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," Sensors-Basel, vol. 17, no. 4, pp. 818, 2017.

[26] S. Kim, J. Kang, M. Lee and S. Song, "DeepFC: ConvLSTM Network for Trajectory Prediction of Tropical Cyclone using Spatiotemporal Atmospheric Simulation Data," 2018.

[27] H. Yao, X. Tang, H. Wei, G. Zheng and Z. Li, Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction, Proc. 2019 AAAI Conference on Artificial Intelligence (AAAI’19), 2019.

[28] L. Y., D. Y., K. W., L. Z. and W. F., "Traffic Flow Prediction With Big Data: A Deep Learning Approach," IEEE Trans. Intell. Transp., vol. 16, no. 2, pp. 865-873, 2015. doi: 10.1109/TITS.2014.2345663.

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

[30] Q. TangM. Yang and Y. Yang, "ST-LSTM: A Deep Learning Approach Combined Spatio-Temporal Features for Short-Term Forecast in Rail Transit," J. Adv. Transport., vol. 2019, 2019.

[31] Y. LiuZ. Liu and R. Jia, "Deep learning based architecture for metro passenger flow prediction," Transportation Research Part C: Emerging Technologies, vol. 101, pp. 18-34, 2019.

[32] J. Li et al., Graph CNNs for Urban Traffic Passenger Flows Prediction, Proc. 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2018, pp. 29-36.

[33] X.M. YaoP. 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, pp. 92-99, 2016.

[34] Han, J. Wang, Ren, Gao and Chen, "Predicting Station-Level Short-Term Passenger Flow in a Citywide Metro Network Using Spatiotemporal Graph Convolutional Neural Networks," International Journal of Geo-Information, vol. 8, pp. 243, 2019, doi: 10.3390/ijgi8060243.

[35] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, Proc. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 770-777.

[36] P. Liu, Y. Zhang, D. Kong and B. Yin, "Improved Spatio-Temporal Residual Networks for Bus Traffic Flow Prediction," Applied Sciences, vol. 9, no. 4, pp. 615, 2019.

[37] J. ZhangY. Zheng and D. Qi, "Deep Spatio-Temporal Residual Network for CITYWIDE CROWD FLOW PREDICTION," Proc. Proceedings of the thirteenth conference on artificial intelligence, 2017.

[38] M.D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, Proc. European conference on computer vision, Springer, 2014, pp. 818-833.

[39] C. Szegedy et al., Going deeper with convolutions, Proc. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[41] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Proc. Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249-256.

[42] K. He, X. Zhang, S. Ren and J. Sun, Identity mappings in deep residual networks, Proc. European conference on computer vision, Springer, 2016, pp. 630-645.

[43] T.N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.
Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P.S. Yu, "A comprehensive survey on graph neural networks," arXiv preprint arXiv:1901.06596, 2019.

G. Li, M. Müller, A. Thabet and B. Ghanem, "Can GCNs Go as Deep as CNNs?" arXiv preprint arXiv:1904.03751, 2019.

A. RahimiT. Cohn and T. Baldwin, "Semi-supervised user geolocation via graph convolutional networks," arXiv preprint arXiv:1804.08049, 2018.

T. Pham, T. Tran, D. Phung and S. Venkatesh, Column networks for collective classification, Proc. Thirty-First AAAI Conference on Artificial Intelligence, 2017.

Q. LiZ. Han and X. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, Proc. Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

L. Y., Q. Y., G. J., C. C., W. Y. and J. L., Short-term forecasting of rail transit passenger flow based on long short-term memory neural network, Proc. 2018 International Conference on Intelligent Rail Transportation (ICIRT), 2018, pp. 1-5.

Z. Zhao, W. Chen, X. Wu, P.C. Chen and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," Iet Intell. Transp. Sy., vol. 11, no. 2, pp. 68-75, 2017.

D. BahdanauK. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

Z. Zhang, M. Li, X. Lin, Y. Wang and F. He, "Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies," Transportation Research Part C: Emerging Technologies, vol. 105, pp. 297-322, 2019.

Q. LiuB. Wang and Y. Zhu, "Short-Term Traffic Speed Forecasting Based on Attention Convolutional Neural Network for Arterials," Computer-Aided Civil and Infrastructure Engineering, vol. 33, no. 11, pp. 999-1016, 2018.

Y. Wu, H. Tan, L. Qin, B. Ran and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," Transportation Research Part C: Emerging Technologies, vol. 90, pp. 166-180, 2018.

Jinlei Zhang was born in Hebei Province, China. He is pursuing a PhD degree in Beijing Jiaotong University. Now he is doing researches in University of Washington as a joint PhD student. His research interests include traffic data mining and applications, dynamic traffic modeling and management.

Feng Chen received his B.S. degree, M.A. degree, Ph.D. degree in railway engineering from Beijing Jiaotong University, Beijing, China, in 1983, 1990 and 2007 respectively. He is now working as a professor in Beijing Jiaotong University and the president of Chang’an University. His research interests include passenger flow management, traffic data mining and application for urban rail transit. Professor Chen won the first prize of national science and technology progress as the second accomplisher in 2017.

Yadi Zhu was born in Henan Province, China. He received the Ph.D. degree from the School of Civil Engineering, Beijing Jiaotong University in 2019. He currently holds a postdoctoral position with the School of Engineering, Center for Advanced Infrastructure and Transportation, Rutgers University.

His research interests are transportation planning, including pedestrian flow modeling, pedestrian simulation research, and transportation demand analysis and forecasting.

Yinan Guo is pursuing a B.S. degree at the Information School of University of Washington. His current research interest lies in the interplay among information science, computer science and applied mathematics. Since high school, Mr. Guo has received one international award and three patents for his invention in technology and research in computer science.